WING FLEXURE COMPENSATION FOR AIRCRAFT ATTITUDE AND POSITION DETERMINATION IN AN INVERTED CARRIER-PHASE POSITIONING SYSTEM

THESIS

Bradley W. Mahlum
Second Lieutenant, USAF

AFIT/GE/ENG/95D-14

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/95D-14

WING FLEXURE COMPENSATION FOR AIRCRAFT ATTITUDE AND POSITION DETERMINATION IN AN INVERTED CARRIER-PHASE POSITIONING SYSTEM

THESIS

Bradley W. Mahlum
Second Lieutenant, USAF

AFIT/GE/ENG/95D-14

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

AFIT/GE/ENG/95D-14

WING FLEXURE COMPENSATION FOR AIRCRAFT ATTITUDE AND POSITION DETERMINATION IN AN INVERTED CARRIER-PHASE POSITIONING SYSTEM

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfilment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Bradley W. Mahlum, B.S. Electrical Engineering

Second Lieutenant, USAF

December 1995

Preface

I would like to take this opportunity to thank a number of people who helped me complete this thesis. First, I would like to thank Captain Ron Delap for all his help as my thesis advisor. Without his encouragement and assistance, this task would have been much more arduous. I truly appreciate all of his support.

I would also like to thank the other members of my committee for their assistance. To Lieutenant Colonel Robert Riggins, my thanks must go out for helping me "see clearly". Thank you to Dr. Meir Pachter for his editing and assistance with information.

Thank you to the Central Inertial Guidance Test Facility (CIGTF) of the 46th Test Group at Holloman AFB, NM for sponsoring this research.

Thank you to my family, the in-laws-to-be, my Grandparents, and most importantly, my parents, who have continuously supported me in my endeavors. Without all the support and love through the years, I would not be where I am. I love each and every one of you and appreciate everything through these twenty-three years.

Finally, thank you to my fiancee, Aimee LaFrance. She has always been understanding of the time I had to take out of visiting her to work on my thesis. Also, she has constantly encouraged me to keep going forward with the work and to keep smiling. For putting up with it all, while completing her nursing degree, I must give my deepest thanks, admiration, and love.

Bradley W. Mahlum

Table of Contents

List of Figures

List of Tables

Abstract

In response to the ever increasing accuracies in inertial navigation systems, the U. S. Air Force must develop higher accuracy reference systems. These reference systems must also be small enough to be utilized in the testing of navigation systems onboard fighter aircraft. One such proposed system utilizes carrier-phase Global Positioning System (CPGPS) transmitters mounted on AIM-9 pods with receivers on the ground. This research examines one possible method of utilizing this system to determine the attitude and position of the aircraft, given position estimates for the transmitter's locations. The transmitter positioning algorithm showed that the geometry will be problematic for this configuration. However, if given the estimates of transmitter positions within the desired accuracy, the aircraft attitude and position algorithms worked effectively.

Wing Flexure Compensation for Aircraft Attitude and Position Determination in an Inverted Carrier-Phase Positioning System

I. *Introduction*

The Central Inertial Guidance Test Facility (CIGTF) at Holloman AFB, New Mexico, is upgrading their Navigation Reference System (NRS) to utilize the precise carrier-phase measurements of the Global Positioning System (GPS). This is a much needed step in order to maintain the necessary accuracy needed to test new navigation systems reliably. Without this advancement, it will be difficult to test accurately the new navigation systems entering the Air Force inventory (1).

The current NRS under development is called CHAPS II, the CIGTF High Accuracy Positioning System. It is comprised of three navigation systems: a Litton LN-93 Inertial Navigation System (INS), a Carrier-Phase Global Positioning System (CPGPS), and a Range/Range-Rate System (RRS) of ground transponders. This system is put into a C-12 type cargo rack and can be flown around the test range calculating the position and velocity of the vehicle utilizing an Extended Kalman Filter (EKF) (2).

CIGTF has now progressed to evaluation of a new system, to be called SARS, the sub-meter accuracy reference system, that will reduce the hardware to two AIM-9 pods, each containing two GPS transmitters, and GPS receivers located on the ground. By utilizing post-processing and known ground locations of the receivers, SARS should

obtain precise attitude and position solutions, while occupying less space on the aircraft than CHAPS II.

1.1 *Background*

CIGTF's current system (CHAPS I) incorporates Differential GPS (DGPS) in the NRS. When CHAPS II development is complete, this will be replaced with Carrier-Phase GPS. CIGTF began progress toward this goal with the efforts, both in the AFIT masters thesis and temporary duty work, of Captain Neil Hansen, Canadian Forces (Air). As the next step, CIGTF sponsored Lieutenant Brian Bohenek's AFIT masters thesis that examined the inclusion of double differencing into the NRS. He is currently assigned to CIGTF working on the SARS development.

By implementing CPGPS in CHAPS II, the NRS will provide a reliable benchmark for testing new navigation systems in the years to come. However, CHAPS II cannot be used in fighter aircraft due to the size of the components involved; the complete system occupies a C-12 cargo rack. Therefore, CIGTF is investigating the possibility of putting GPS transmitters onto AIM-9 pods which can be mounted on the wings of fighter aircraft in order to test these navigation systems. It is important to note that this system eliminates the use of an INS and data recorders on the aircraft, which greatly reduces the on-board space requirement (3).

1.2 Problem Statement

While AIM-9 pods on wings of fighter aircraft will provide a highly flexible system, the flexure of the wing will introduce difficulties in locating the point on the plane where the navigation system under test is located. In order to allow precise position and attitude determination, a method of modeling and/or filtering these flexures will be necessary. This research will develop an attitude determination algorithm for this configuration and will investigate methods of compensating for the flexure effects on the solution.

1.3 Current Research

Although no experiments like this have been conducted yet, several preliminary attitude and position determination experiments have been conducted. In addition, Dr. Frank van Graas and David Diggle of Ohio University, and Richard Hueschen of the NASA Langley Research Center are working on an application of CPGPS to precision landing systems (4). Meanwhile, Clark Cohen and Bradford Parkinson of Stanford University, and David McNally of the NASA Ames Research Center are conducting experiments using CPGPS to determine the attitude of an aircraft (5).

Dr. van Graas and associates implemented a real-time attitude and heading determination system using CPGPS. As testing continued, they realized that it would be possible to incorporate this system in a landing system and eventually as a flight reference system (FRS). This experimentation had several main goals: a 0.1 meter spherical error probability (SEP) accuracy, multiple updates per second, and repeatability of flight paths

within a couple of meters. These goals were set for several reasons. The primary need for the goals was accuracy of less than 1 foot SEP in order to be used as a landing system for the most stringent instrument landing categories. Another need was the ability of the system to self-evaluate its performance. During testing, Dr. van Graas and associates considered seventeen flights for statistical analysis, adequate to meet the requirements of Monte Carlo calculations. The results of this analysis were as follows: the SEP was approximately 21 centimeters or 0.69 feet with a one-sigma value of approximately 17 centimeters or 0.5 feet. These results showed that the CPGPS was capable of obtaining accuracies on the order of 10 centimeters when everything functioned properly, a key issue in CPGPS. Currently, they are working on ways to improve this even further. This experiment demonstrates that CPGPS is certainly a viable option for landing systems (4).

While Dr. van Graas was conducting the above tests, Dr. Cohen and associates were conducting tests into attitude determination utilizing CPGPS. Attitude determination is concerned with describing the orientation of an aircraft relative to the horizon or to the navigation frame. In order to perform this test, multiple receivers are needed at varying locations on the aircraft. Dr. Cohen and associates placed their four receivers on the wingtips, on the tail of the aircraft, and at the center of the aircraft. Each of these receivers functions as though it were independent, arriving at a solution of its position in space. A central computer then takes all of these measurements, and based upon a structural model of the aircraft and the receiver spacing, determines each receiver's relative position as compared to the other three receivers. From this solution, the attitude of the aircraft is found.

Several problems arise when considering such an arrangement. The most significant problem is called masking. When masking occurs, a part of the aircraft obscures the antenna's line of sight to the satellites, making it difficult to receive the signals. Loss of satellites for determining position may occur. Once satellites are lost, the lengthy reacquisition process must be accomplished before solutions can again be output. Another problem is the potential for too few satellites to be in view; not enough information is available for the receivers to obtain a solution. Finally, the aircraft itself flexes slightly under the forces it encounters, a particularly important aspect for this research. This motion can cause wingtips to sway as much as several feet on a B-52. When this occurs, the orientation of the receivers is no longer valid for attitude determination, only approximate.

After testing was complete, Dr. Cohen made several observations. The most important was that the CPGPS attitude was still performing accurately during periods of shading or satellite shortages. The attitude determined through CPGPS was never more than 0.2 degrees, in pitch, roll, or yaw, off of the true attitude. Also, it still performed adequately under extreme wing flexure. These results proved the viability of attitude determination through CPGPS (5).

CIGTF is currently sponsoring research into resolution of other problems that need to be resolved in order to fully implement the proposed method. These include methods of eliminating the cycle slip problems with CPGPS and resolution of position accuracies to the ten centimeter level.

Current research into wing flexure is focusing on adaptive wing structures (6). These so called "smart" structures include lift and control surfaces that can be computer adjusted. This allows for elastic response to reduce forces on the wing structure providing more aircraft stability. As a result of this variability and enhanced stability, icing and wing damage effects could be minimized (6). This would provide a potential benefit for SARS in that the computer controlling these surfaces could store its data, allowing for more precise knowledge of the wing's actual flexure relative to the aircraft during the flight, resulting in more accurate estimations of the aircraft's position.

1.4 Assumptions

As a matter of convenience for readers, the major assumptions included in this research will be described in this section. This should allow the reader to determine the research's appropriateness for his/her application.

1. All results are generated from computer simulations of the system. It will be necessary to conduct further experimentation with actual flight tests.
2. The wing flexure models provided by the F-16 C/D Block 40 Aircraft Wing Twist Analysis Report (7) are assumed to be correct and will not be further investigated for improvement.
3. Cycle slips and phase ambiguities will not be dealt with in this research. Although these are significant problems in resolving position, several methods exist to deal with them, and this research will neither redevelop nor recreate these (3).

4. The ground receivers' locations are exactly known. After self-surveying by the receivers, the accuracies should be sub-centimeter (8).
5. The aircraft to be simulated is an F-16. Although wing flexure models will not be the same as for those of other aircraft, it was felt this represented a good starting point (3). Only minor modifications would be required to simulate other wing flexure models.
6. The system will not be subject to masking, or shadowing, due to the location of the transmitters. Since the transmitters are on the wingtips, full coverage should be available, except for the singular case when the aircraft is at 90^{o}, pitch or roll. Even in these cases, receivers down range should be able to detect these signals.
7. The transmitters are assumed to be 9 feet apart on the AIM-9 pods. This baseline represents the distance from front to rear of the AIM-9 pod (9).
8. The flight profile is restricted to Mach 0.6 and turns not exceeding 6 g's, where $1\ g = 9.8\ m/s^2 = 32\ ft/s^2$.
9. A spherical earth was assumed for these simulations. This was merely to make aircraft movement an easier task in the computer. This assumption does not damage in any way the results, since this research was desired as a proof of concept for CIGTF (3).

1.5 Scope

This thesis will focus on methods of compensating for the wing flexure. Once a rigid body algorithm for position and attitude determination is operational, it will be expanded to account for the flex of the wings providing a location of the center of the

aircraft. CPGPS solutions will be evaluated. Also to be evaluated is the efficiency of the position and attitude determination algorithms developed in this research.

1.6 Approach/Methodology

This research will examine models for wing flexure in an effort to account for its effects on the navigation solution provided by the transmitters. In order to do this, the following steps will be employed:

1. An algorithm for position and attitude determination based upon a rigid body assumption will be developed first. This will provide a good starting point to build further algorithms accounting for all necessary effects.
2. Algorithms simulating the flexure of the wing's effects on transmitter locations will be developed.
3. Methods of solving for the aircraft position and attitude will be developed and evaluated to recommend the most efficient method.

1.7 Overview of Thesis

This thesis contains five chapters. Chapter I presents the problem to be solved and the methods that will be employed in order to solve this problem. Chapter II describes the theory necessary to complete this thesis; including background on carrier-phase measurements, differencing techniques, and wingtip oscillation models. Chapter III discusses models and algorithms developed to estimate position and attitude. The results are provided in Chapter IV, and Chapter V details conclusions and recommendations.

II. Theory

2.1 Overview

This chapter provides an overview of theory for Carrier-Phase GPS and wingtip oscillations. In the Carrier-Phase GPS section, observation and measurement equations are explained, as well as error sources in the measurement equation. Also, differencing techniques to help reduce these errors are explained. In the wingtip oscillation section, a basic background is provided.

For further information on CPGPS, reviewing references (1, 2, 10, 11, 12) is recommended. Further information on wing twisting can be obtained by reading (7) or contacting (13).

2.2 Carrier-Phase Global Positioning System Measurements

2.2.1 Carrier-Phase GPS Observation Equations. The carrier-phase observation equation represents the process by which a measurement is generated. A measurement results from subtracting a receiver-generated carrier signal from the GPS satellite transmitted carrier signal that the receiver tracks. This result is referred to as the carrier beat phase observable and is represented by the following equation

$$\phi = \phi^{k}(T) - \phi_{i}(t) \tag{2.1}$$

where ϕ is the carrier beat phase observable, T represents the time of transmission, based on the satellite clock, $\phi^{k}(T)$ represents the phase of the transmitted carrier signal from

the k^{th} satellite at time T, t is the time of reception of the signal, based on the receiver clock, and $\phi_i(t)$ represents the phase of the i^{th} receiver's internally generated signal at time t. For the rest of this thesis, the i and k symbols will be dropped and T and t will be used to differentiate which phase term is being referenced. The transmitted signal's time of travel, δ t, is:

$$\delta \text{ t} = \text{t} - \text{T} \tag{2.2}$$

so that

$$\phi(\text{t}) = \phi(\text{T} + \delta \text{ t}) \tag{2.3}$$

Now using a Taylor series expansion about T and assuming δ t is small leaves:

$$\phi(\text{T} + \delta \text{ t}) = \phi(\text{T}) + \left.\frac{\partial \phi(\tau)}{\partial \tau}\right|_{\tau=T} \cdot \delta \text{ t} + higher.order.terms \tag{2.4}$$

The first derivative of $\phi(T)$ with respect to time equals the frequency, f. Since the GPS phase/frequency relationship is valid for very stable oscillators over short time intervals, the higher order terms can be considered negligible. Therefore, the equation, with frequency assumed constant, can be simplified to:

$$\phi(\text{T} + \delta \text{ t}) = \phi(\text{T}) + \text{f} \cdot \delta \text{ t} \tag{2.5}$$

Combining Equation (2.3) with Equation (2.5), this becomes:

$$\phi(\text{t}) = \phi(\text{T}) + \text{f} \cdot (\text{t} - \text{T}) \tag{2.6}$$

which can be substituted in Equation (2.1) to form:

$$\phi = \phi(T) - \phi(t) = -f \cdot (t - T) \tag{2.7}$$

To relate the transmission and reception times, T and t, note that the satellite and receiver clocks are independent. Also, each clock is slightly off true GPS time by some error. If true GPS time is indicated as t_{GPS}, reception time can be related by:

$$t_{GPS}=t+dt \tag{2.8}$$

where dt is the receiver clock offset from true GPS time. The transmission time, T, can be similarly expressed, using the satellite clock offset from true GPS time, dT. The propagation time should equal the difference of these measurements, which is also equal to the distance traveled, with errors included, divided by the speed of light, as seen in the following expression:

$$T_{propagation} = \frac{(\rho - d_{ion} + d_{trop})}{c} \tag{2.9}$$

where ρ is the true range from the satellite to the receiver, d_{ion} is the distance equivalent of the ionospheric offset, and d_{trop} is the distance equivalent of the tropospheric delay. The minus sign on the ionospheric delay term represents the phase advancing effect of the ionosphere. Relating this equation to the true GPS time, Equation (2.8), yields:

$$t_{GPS} = T + dT + \frac{(\rho - d_{ion} + d_{trop})}{c} \tag{2.10}$$

Now setting Equations (2.8) and (2.10) equal to each other provides the relation between the transmission and the reception times:

$$T + dT + \frac{(\rho - d_{ion} + d_{trop})}{c} = t + dt \tag{2.11}$$

and rearranging terms yields:

$$t - T = dT - dt + \frac{(\rho - d_{ion} + d_{trop})}{c} \tag{2.12}$$

The observation equation can then be obtained by substituting this equation into Equation (2.7) leaving:

$$\phi = -f \cdot (dT - dt) - \frac{f}{c} \cdot (\rho - d_{ion} + d_{trop}) \tag{2.13}$$

From this equation, it is easily seen that the carrier phase observable is a function of the frequency of the transmitted signal, the satellite and receiver clock errors, the range between satellite and receiver, and the atmospheric delays.

2.2.2 Carrier-Phase GPS Phase Range Measurement Equations. The carrier phase measurement only measures the phase shift between the satellite and receiver generated carrier signals. As such, it only accounts for a fraction of the total wavelengths between satellite and receiver. The total number of phase cycles, at time t, can be represented as:

$$\phi_{total}(t) = \phi_{frac}(t) + \phi_{int}(t_o, t) + N(t_o) \tag{2.14}$$

where $\phi_{frac}(t)$ is the fractional part of the total wavelength that is measured, $\phi_{int}(t_o, t)$ is an integer number of phase cycles from the initial time, t_o, to the time considered, t, and $N(t_o)$ is the phase ambiguity term. Figure 2.1 illustrates each part of the total number of cycles.

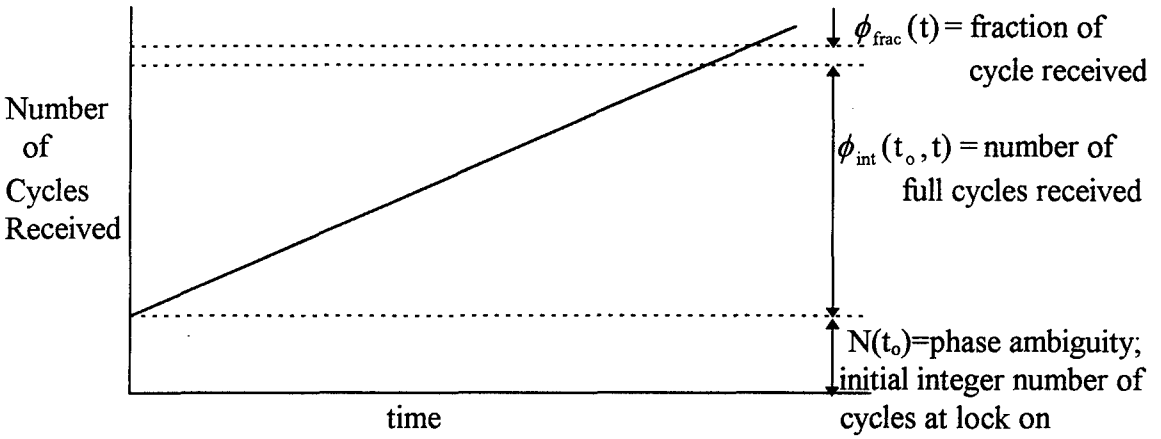

Figure 2.1 Illustration of components of total phase

Note that the solid line represents the portion of the number of cycles that the receiver is measuring. Phase ambiguity, also referred to as the cycle ambiguity, is the difference between the true number of phase cycles between satellite and receiver and the integer count currently calculated at the receiver, at the initial time or just after each reacquisition. As long as the receiver keeps lock on the satellite, this ambiguity term remains constant, but if a cycle slip occurs the ambiguity term will change. Since these slips occur frequently, the phase ambiguity is a time-varying number.

The carrier phase measured by the receiver is the fractional phase portion at time t and the integer portion at time t, represented by:

$$\phi_{measured}(t) = \phi_{frac}(t) + \phi_{int}(t_o, t) \tag{2.15}$$

The total phase cycle at time t can be expressed as:

$$\phi_{total}(t) = \phi_{measured}(t) + N(t) \tag{2.16}$$

By substitution of the phase observation equation into this equation, the phase-range for the carrier-phase observable is formed:

$$\phi_{measured}(t) = -f \cdot (dT - dt) - \frac{f}{c} \cdot (\rho - d_{ion} + d_{trop}) - N(t) \tag{2.17}$$

This equation is the measured phase range expressed as a number of carrier cycles. For positioning applications, this number needs to be expressed in units of length. In order to convert from cycles to length, multiply by the negative of the carrier wavelength to get positive length units. This is necessary due to the fractional portion being measured as a negative number. The resulting equations are:

$$\begin{aligned} \Phi(t) &= -\lambda \cdot \phi_{measured}(t) \\ \Phi(t) &= \rho + c \cdot (dt - dT) + \lambda \cdot N(t) - d_{ion} + d_{trop} \end{aligned} \tag{2.18}$$

Here, the Φ in the equations represents the measured range from a satellite to a receiver in units of length, λ in the equations is the wavelength of the carrier phase signal, and ϕ is the measured number of carrier cycles between the transmitter and the receiver. This equation provides a phase range that is analogous to the pseudorange in standard GPS operation.

2.2.3 Differencing Techniques. Equation (2.18) illustrates the measured phase range with all of its error terms. These error terms need to be reduced in order to obtain more accurate solutions. One method of reducing, or eliminating, these errors is through differencing. There are three basic differencing methods, referred to as single differencing techniques: between-epoch, between-receiver, and between-satellite. To differentiate these, the following symbols are implemented: δ for between-epoch differences, Δ for between-receiver differences, and ∇ for between-satellite differences. It is also possible to combine these methods into a double difference and even higher order differences. This section will cover the single differencing techniques and double differencing techniques.

2.2.3.1 Single Differencing. As previously stated, there are three single differencing forms. The between-epoch single difference takes the difference of phase range measurements between two time epochs, t_1 and t_2. To apply this method, the difference must be made between the same satellite/receiver pair. Figure 2.2 demonstrates this method pictorially.

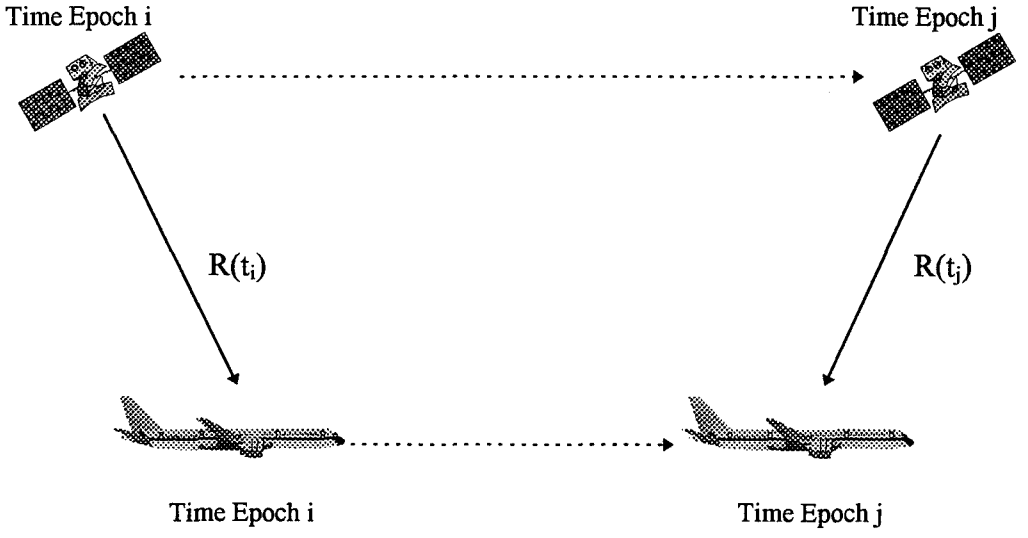

Figure 2.2 Figure Depicting CPGPS Between-Epoch Single Difference:
$\delta R(t_i - t_j) = R(t_i) - R(t_j)$

At the ith epoch, the phase-range can be written as:

$$R(t_i) = R_t(t_i) + \delta R_{uclk}(t_i) + \delta R_{sclk}(t_i) - \delta R_{ion}(t_i) + \delta R_{trop}(t_i) + \delta R_N(t_i) + v(t_i) \quad (2.19)$$

where: $R(t_i)$ = phase range measurement at time t_i

$R_t(t_i)$ = true range between satellite and receiver at time t_i

$\delta R_{uclk}(t_i)$ = range error due to receiver (user) clock error at time t_i

$\delta R_{sclk}(t_i)$ = range error due to satellite clock error at time t_i

$\delta R_{ion}(t_i)$ = range error due to ionospheric delay at time t_i

$\delta R_{trop}(t_i)$ = range error due to tropospheric delay at time t_i

$\delta R_N(t_i)$ = range error due to phase ambiguity at time t_i

v_i = measurement noise

Subtracting the phase-range measurements between the ith and jth epochs forms the single difference:

$$\delta_t R(t_i - t_j) = \delta_t R_t(t_i - t_j) + \delta_t \delta R_{uclk}(t_i - t_j) + \delta_t \delta R_{sclk}(t_i - t_j) - \delta_t \delta R_{ion}(t_i - t_j) + \delta_t \delta R_{trop}(t_i - t_j) + \delta_t v \quad (2.20)$$

If no cycle slips occur between the times t_i and t_j, the phase ambiguity term remains constant, and examination of this equation shows that the phase ambiguity term is then eliminated as a source of error. If a cycle slip does occur, a term equal to the range equivalent of the number of slips would be added in here. Also note that the receiver and/or satellite clock errors would be eliminated if either or both were time-invariant.

A between-receiver single difference is the difference in phase-range measurements between two receivers tracking the same satellite. Figure 2.3 illustrates this method. The phase-range measurement at the ith receiver is described by:

$$R_i = R_{ti} + \delta R_{uclki} + \delta R_{sclk} - \delta R_{ioni} + \delta R_{tropi} + \delta R_{Ni} + v_i \qquad (2.21)$$

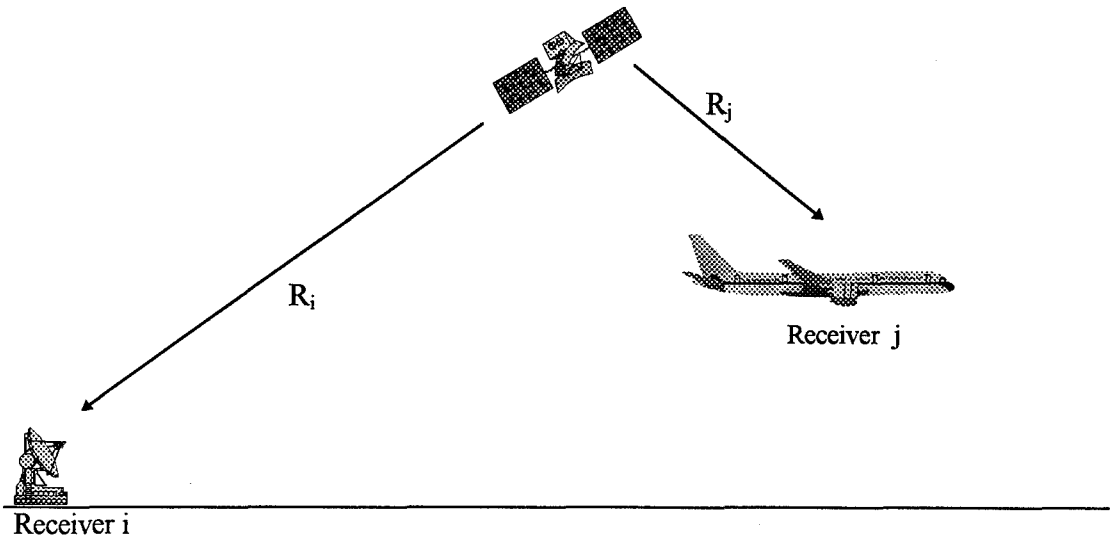

Figure 2.3 Figure Depicting CPGPS Between-Receivers Single Difference: $\Delta R_{ij} = R_i - R_j$

where this equation utilizes the standard notation from before and the i subscript indicates the ith receiver. Subtracting the phase-range of the jth receiver from that of the ith receiver leaves the between-receiver single difference. This method eliminates the range equivalent of the satellite clock error, since the same satellite is being used by each receiver. This can be mathematically expressed as:

$$\Delta R_{ij} = \Delta R_{tij} + \Delta\delta\ R_{uclkij} - \Delta\delta R_{ionij} + \Delta\delta R_{tropij} + \Delta\delta R_{Nij} + \Delta v_{ij} \qquad (2.22)$$

This differencing yields an expression for the relative difference between the two receivers and therefore helps to reduce the atmospheric delay terms as long as the receivers are relatively close. If one receiver is at a fixed, surveyed point and both receivers are tracking the same set of satellites, this technique is referred to as Differential GPS.

Between-satellites single differences are basically the reverse of the between-receivers differences; the difference is between two satellite phase-range measurements to the same receiver. Figure 2.4 illustrates the between-satellites single difference.

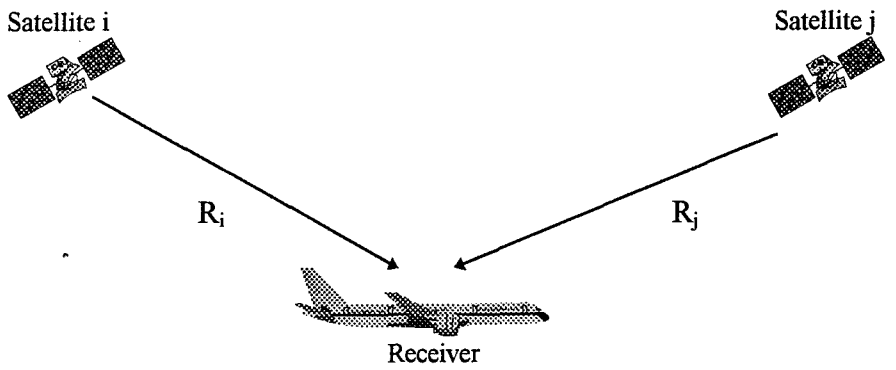

Figure 2.4 Between-Satellites Single Difference: $\nabla R_{ij} = R_i - R_j$

Using the previous notation, the phase-range measurement from the ith satellite to the receiver is:

$$R_i = R_{ti} + \delta R_{uclk} + \delta R_{sclki} - \delta R_{ioni} + \delta R_{tropi} + \delta R_{Ni} + v_i \qquad (2.23)$$

Now, taking the difference of the phase-ranges between the ith and jth satellites yields:

$$\nabla R_{ij} = \nabla R_{tij} + \nabla \delta R_{sclkij} - \nabla \delta R_{ionij} + \nabla \delta R_{tropij} + \nabla \delta R_{Nij} + \nabla v_{ij} \qquad (2.24)$$

This equation represents the relative distance between the satellites. The user clock error term is eliminated by this difference.

2.2.3.2 Double Differencing. Subtracting two different single differenced measurements described in Section 2.2.3.1 creates a double difference measurement. There are three different double differencing methods: between-receivers/epochs, between-satellites/epochs, and between-receivers/satellites. A between-receivers/epochs double difference takes the between-receivers single difference measurement at one epoch and differences it with a between-receivers single difference from a later epoch. This technique yields the equation:

$$\begin{aligned}\delta_t \Delta R(t_i - t_j) = \delta_t \Delta\delta R_{ti}(t_i - t_j) + \delta_t \Delta\delta R_{uclki}(t_i - t_j) - \delta_t \Delta\delta R_{ioni}(t_i - t_j) \\ + \delta_t \Delta\delta R_{tropi}(t_i - t_j) + \delta_t \Delta v_i\end{aligned} \tag{2.25}$$

NOTE: the symbol $\delta_t \Delta$ reflects the between-receivers single difference combined with the between-epochs single difference. This eliminates the satellite clock error and the phase ambiguity term, as long as no cycle slips have occurred. Also, it further reduces the atmospheric delay errors.

The between-satellites/epochs double difference takes the difference of a between-satellites single difference at a epoch and again at a later epoch. Mathematically, this can be expressed as:

$$\begin{aligned}\delta_t \nabla R(t_i - t_j) = \delta_t \nabla\delta R_{ti}(t_i - t_j) + \delta_t \nabla\delta R_{sclki}(t_i - t_j) - \delta_t \nabla\delta R_{ioni}(t_i - t_j) \\ + \delta_t \nabla\delta R_{tropi}(t_i - t_j) + \delta_t \nabla\delta v_i\end{aligned} \tag{2.26}$$

This technique eliminates the user clock bias and the satellite clock bias, as long as no clock drift is present, and the phase ambiguity term, as long as no cycle slips occur.

The between-receivers/satellites double difference is the most useful for general application. Due to the implementation, it eliminates the most error terms and is relatively easy to use. This method subtracts a between-receiver single difference from another between-receiver single difference using the same set of receivers and a different satellite. Figure 2.5 is provided to help clarify the procedure. This difference results in the following equation:

$$\nabla\Delta R_{ij} = \nabla\Delta R_{tij} - \nabla\Delta\delta R_{ionij} + \nabla\Delta\delta R_{tropij} + \nabla\Delta\delta R_{Nij} + \nabla\Delta v_{ij} \qquad (2.27)$$

where i and j are the two satellites from which measurements are taken. As shown in the equation, the satellite and user clock error terms are eliminated. Also, the atmospheric error terms are reduced significantly.

2.2.4 Cycle Slips. Although cycle slips will not be covered as part of this research, it is necessary to understand what a slip is. A cycle slip occurs when there is a loss of lock between receiver and satellite. If this occurs, the receiver will try to regain the signal from the satellite. While this is occurring, the receiver will not count the phase cycles and some will go uncounted, referred to as "slipped." When the receiver regains lock, it will start counting phase cycles as though it never lost lock. This may change the integer portion of the measurement without any indication, producing a difference in the phase-range measurement which is equal to the number of slipped cycles. Therefore, the phase ambiguity term would have to be adjusted.

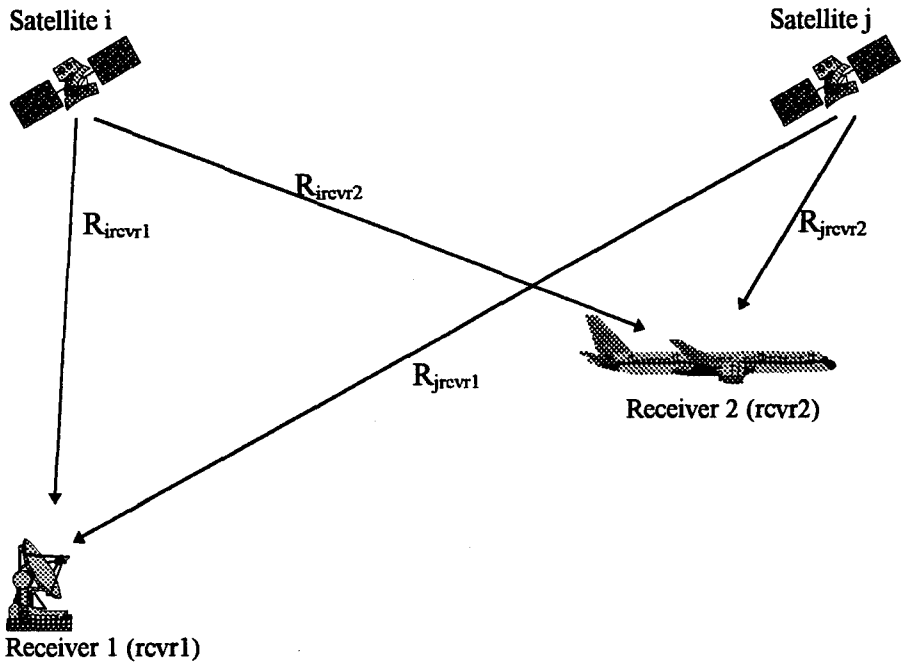

Figure 2.5 Between-Receivers/Satellites Double Difference:
$\nabla\Delta R_{ij} = \Delta R_{i\,rcvr1,rcvr2} - \Delta R_{j\,rcvr1,rcvr2}$

Cycle slips are unpredictable due to the variety of causes. High dynamic maneuvers, weather, noise interference, natural obstacles, and jamming are primary causes.

2.3 Wing twisting. An important consideration in precise positioning of this nature is the behavior of the AIM-9 pods mounted on the wingtips. This behavior is affected by wing twisting. Recent studies into this behavior have focused on the "smart" structures described in Section 1.3.

In 1988, the Air Force conducted a study of the AIM-9 pod's relative orientation to the body. Most of this section discusses this study of Block 40 F-16s (7).

Since air-to-air missiles need to establish a "lock" on a target, the missile needs to establish the position of the target. The initial acquisition is through the fire control radar, leading to the necessity of knowing the relative angles of the missile with respect to the aircraft. Static relative alignment is easily determined; for the F-16, this initial alignment is pod nose down three degrees (see Figure 2.6). During flight, this will change due to flexure of

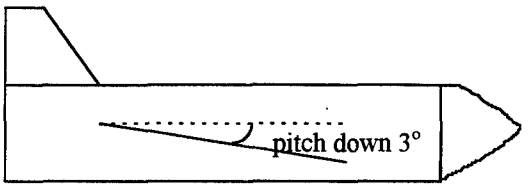

Figure 2.6 Illustration of Initial Pod Pitch

the wing, which will be referred to as "wing twist." Primary factors in the wing twist are Mach number, altitude, and load factor. Other factors include the location of the center of gravity and the gross weight.

In the experiment conducted, scale models were utilized to develop wing pressure measurements which were combined with the airplane's flexibility matrix. This was conducted for a variety of Mach numbers and angle of attacks. The results were then used to generate graphs of the relative angle of the AIM-9 pod to the fuselage.

2.4 Summary. This chapter discussed the basic theory necessary for understanding this research. Carrier-phase GPS was discussed, including the measurement equations utilized by CPGPS and several differencing techniques. Also explained was the

report on F-16 wing flexure. With this knowledge, the models that will be used in this research will be discussed.

III. Models

3.1 Overview

This chapter covers the models utilized in this thesis. The first part discusses the reference model implementation, while the second part covers the solution technique chosen to explore in this research. Only the actual implementations and justifications will be discussed. The theory behind this research is covered in Chapter 2.

3.2 Reference Model. In order to get the most accurate solutions, it is essential to develop the most complete reference model. While it is desirable to include every conceivable state in the reference model, it is also essential that the model be simple enough to implement. The states considered for this thesis and those that were excluded will be developed in the rest of this section. Those states left out would provide possible ideas for future research.

3.2.1 Aircraft Motion. The first essential step in developing a reference model is to model the aircraft's motion. To do so, one must understand the definitions of heading, pitch, and roll. Figures 3.1 through 3.3 illustrate these concepts. Please note that the arrows indicate the direction of rotation for a positive angle. Also, in Figure 3.3, the direction of aircraft travel is "into the page".

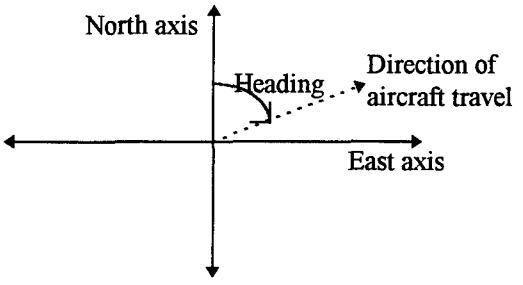

Figure 3.1 Illustration of Heading - Viewed From Above

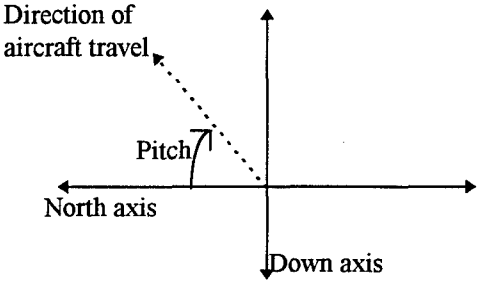

Figure 3.2 Illustration of Pitch

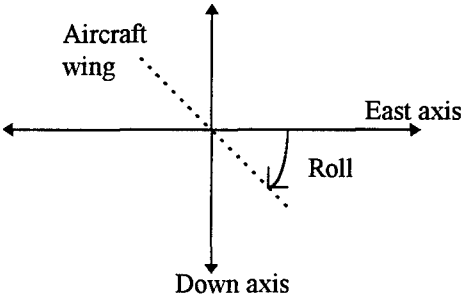

Figure 3.3 Illustration of Roll

For simplicity, it is assumed that any pitch up or down represents a climb or a dive on the part of the aircraft. This is not actually true with real world aircraft. During level flight all aircraft have a small amount of pitch. This pitch is speed dependent and airframe dependent; as the aircraft's speed decreases the pitch to keep the aircraft in level flight increases to a limiting value. This speed dependent pitch is not modeled for this research

as it should not significantly impact the results, but could be utilized in future research reference models. With this assumption, and assuming measurements are taken at one second intervals, the climb or dive rate in feet per second is easily represented by the sine of the pitch angle multiplied by the speed of the plane. A negative pitch indicates a dive, while positive pitch (as shown in Figure 3.2) represents climb.

To move the aircraft in latitude or longitude is slightly more problematic. The heading angle is used to determine how much to move the aircraft in the east or north directions, and is referenced to due north. A positive heading is clockwise from north, while a negative heading is counterclockwise from north. Figure 3.1 clearly shows that the displacement in the north direction is represented by the speed of the plane multiplied by the cosine of the heading, with no pitch. Similarly the movement in the east direction is represented by the speed of the plane multiplied by the sine of the heading, with no pitch. When pitch is included, it is necessary to take the speed in the down direction out of the total speed and then perform this breakdown. This, however, does not represent the final solution. Since the latitude and longitude coordinates are in degrees, the displacement must be represented in terms of degrees, not feet. Therefore, it is necessary to create a routine to convert from feet to degrees. This is easily done by utilizing the geometric relations of the arc length to the angle of the arc. Stated simply, the ratio of the arc length to the circumference of the circle is the same as the ratio of the angle of the arc to 2π or 360 degrees, and Equation 3.1 shows this relation (14).

$$\frac{\text{Distance Traveled in ft.}}{(r_e + \text{aircraft alt}) \times 2\pi} = \frac{\text{Distance Traveled in degrees}}{360^0} \qquad (3.1)$$

Note that the denominator representing the circumference is calculated using the radius of the earth, r_e, plus the altitude of the aircraft. The value of r_e for this thesis is 20925696 feet, from the world geodetic survey of 1984, assuming a spherical earth (15). Including the aircraft altitude is necessary for improved accuracy. The ratio in Equation (3.1) yields the degrees of aircraft movement in the north and east directions and allows the simple addition of these to the old latitude and longitude coordinates to find the aircraft's current position.

3.2.2 Wingtip Transmitter Motion. Once the coordinates of the aircraft are calculated, it is necessary to calculate the coordinates of the GPS transmitters on the wingtips. The simplest possible case of this would be when the wings are rigid. In this case the transmitters would lie along body frame coordinates that would be zero in the z direction, and would have fixed values in the x and y directions. Figure 3.4 shows a top down view and the corresponding x and y body frame coordinates of the transmitters.

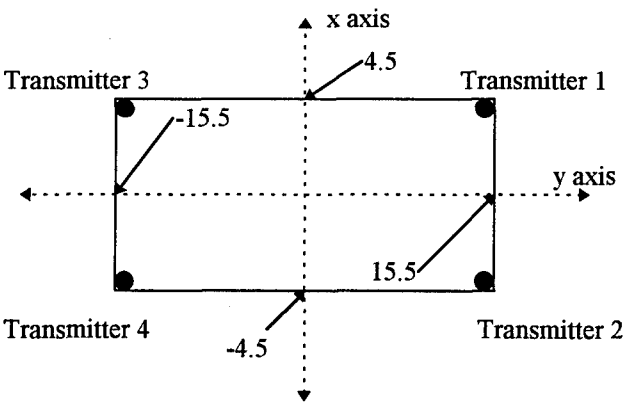

Figure 3.4 Top-Down View of Aircraft Wing with Transmitters in Body-Frame Coordinates

Note that transmitter one is in the positive x, positive y location while transmitter two is in the negative x, positive y location. Under the rigid wing conditions, if the location of transmitters one through four can be determined, then the aircraft motion can be computed using the principles of Section 3.2.1. For example, transmitters one and two could be used to determine aircraft pitch. However, aircraft wings cannot be made rigid because problems with the structural integrity of the wing would occur as the forces acting on it became too great. Therefore, all wings undergo some bending, which must be modeled.

The F-16 C/D Block 40 Aircraft Wing Twist Analysis Report (7) done for the F-16 System Program Office, Wing Division, describes the relative angles of wingtip-mounted AIM-9 pods to an F-16 aircraft. This report formed the basis of the reference model for the wingtip bending, as these are the primary effects in wing flexure (13). The report shows that only the roll and pitch of the wingtip pods need to be considered. See Figures 3.5 and 3.6 for illustrations of pod roll and pitch. The wingtip AIM-9 pods are considered in this thesis as the method for mounting the GPS transmitters at the front and back of the pod. Using the data provided by the report, a Matlab (16) function was developed to determine the roll and pitch of the pod by interpolating over a basic

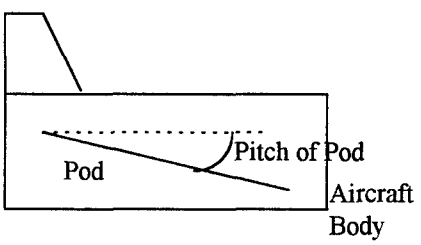

Figure 3.5 Pictorial Representation of Pod Pitch

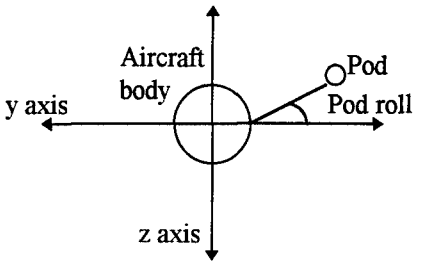

Figure 3.6 Pictorial Representation of Pod Roll

table of the values at nominal points. The data used for pitch are summarized in Table 3.1 and the data used for roll are summarized in Table 3.2.

Mach Number	Altitude in feet				
	Sea Level	4000	10000	25000	50000
0.00	-0.10	-0.10	-0.10	-0.10	-0.10
0.60	-0.75	-0.60	-0.50	-0.25	-0.075
0.85	-2.00	-1.70	-1.40	-0.80	-0.20
0.90	-2.25	-2.00	-1.65	-0.775	-0.25
0.95	-4.05	-3.50	-3.00	-1.40	-0.60
1.10	-3.00	-2.55	-2.10	-1.00	-0.80
1.20	-3.20	-2.60	-2.05	-1.10	-0.60
1.40	-2.90	-2.475	-1.95	-1.00	-0.60
1.50	-2.80	-2.45	-1.90	-0.95	-0.50

Table 3.1 Pitch (in degrees) of AIM-9 Pod Data based on Altitude and Mach Number

It is important to note that for this research, the Mach number calculation utilized the nominal speed of sound in air instead of the formula that calculates the speed as a function of pressure. While the formula would be more accurate, the accuracy gain was not sufficient to justify its use in this research. For more information on the pressure adjustment of the speed of sound, and its impact, refer to (17).

Mach Number	Altitude in feet				
	Sea Level	4000	10000	25000	50000
0.00	-0.70	-0.80	-0.80	-0.80	-0.80
0.60	-0.20	0.00	0.20	0.50	0.50
0.85	-1.50	-1.10	-0.50	0.10	0.80
0.90	-2.00	-1.30	-0.90	0.20	0.90
0.95	-4.00	-3.00	-1.75	0.25	1.10
1.10	-7.10	-5.95	-4.30	-1.10	0.80
1.20	-7.00	-5.50	-3.95	-1.25	0.30
1.40	-8.90	-7.05	-5.00	-1.90	-0.10
1.50	-9.00	-7.20	-5.30	-2.10	-0.75

Table 3.2 Roll (in degrees) of AIM-9 Pod Data based on Altitude and Mach Number

Given the relative angles of the pod to the aircraft, transformations were developed to determine the body-frame coordinates of the transmitters. Once the body frame coordinates have been determined, navigation frame coordinates are computed via the direction cosine matrix given in Equation 3.2 (18). In this matrix, ψ is the heading, θ is the pitch, and ϕ is the roll of the aircraft.

$$C_b^n = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \quad (3.2)$$

This navigation frame representation is then converted to earth-centered, earth-fixed (ecef) coordinates via the direction cosine matrix given in Equation 3.3 (19). In this

$$C_n^e = \begin{bmatrix} -\sin L\cos l & -\sin l & -\cos L\cos l \\ -\sin L\sin l & \cos l & -\cos L\sin l \\ \cos L & 0 & -\sin L \end{bmatrix} \quad (3.3)$$

matrix, L represents the latitude and l is the longitude. This step yields a vector, of the desired length and direction, but centered at the origin of the ecef frame. These ecef coordinates representing the vector of this magnitude can be added to the ecef coordinates of the center of the aircraft to get the correct transmitter locations in ecef. It is important to note that ecef coordinates are being used to simplify the carrier-phase simulation computations.

3.2.3 Carrier-Phase Measurement Simulation. Once all the coordinates are calculated, the carrier-phase measurements can be generated. Using the earth-centered, earth-fixed coordinates of the transmitters and the receivers, a standard distance calculation, shown in Equation 3.4 for the distance between transmitter one (xmtr1) and receiver one (rcvr1) finds the true range between them (14).

$$\text{true range} = \sqrt{\left((x_{rcvr1} - x_{xmtr1})^2 + (y_{rcvr1} - y_{xmtr1})^2 + (z_{rcvr1} - z_{xmtr1})^2\right)} \qquad (3.4)$$

It is then necessary to add the error terms associated with a carrier-phase measurement to this range. Since the plane will be flying in the earth's atmosphere at altitudes lower than the ionosphere, the ionospheric error term from the full carrier-phase equation in Chapter 2 can be eliminated. Also, due to the short duration of the experiment, and the stable nature of the oscillators in the clocks, it is possible to eliminate clock drift errors, leaving clock biases, tropospheric errors, and measurement noises as the only error sources. To represent these, typical error magnitudes (20) are used and added to the range calculated

by Equation 3.4. In order to account for the noise, a zero mean Gaussian random number with a variance of one foot was included. This is actually larger than measurement noise usually included in CPGPS simulations in order to account for the small noise terms in the wing modeling. This results in simulated carrier-phase measurements for all times and all transmitter/receiver combinations and led to the proposed solution model/algorithm.

3.3 Proposed Solution.

3.3.1 Transmitter Positions. The most essential step in implementing a solution to determine aircraft attitude and position is to calculate the locations in space of the transmitters. In order to do this, a least squares positioning algorithm, shown in Figure 3.7, is implemented (21, 22).

The least squares algorithm begins with a nominal position and user clock bias "guess", represented by the vector $\overline{v_{nom}}$, and then calculates an H matrix, given in Equation 3.5, based on that nominal position. Note that in the guess, a term estimating the clock bias is also included as v_{nom4} in order to account for the bias. The term pr_{rcvri} represents the phase range to the ith receiver. Hence, this finds a solution of transmitter position based upon four receivers.

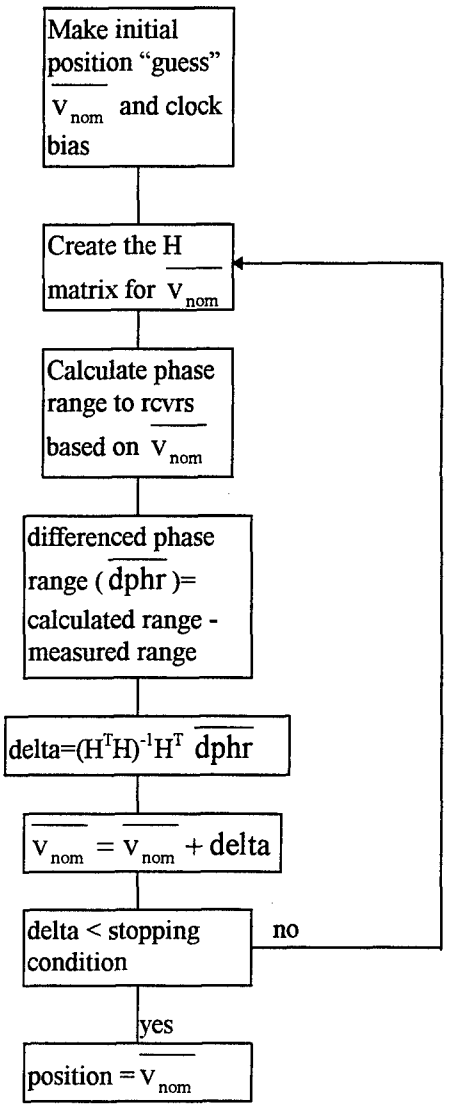

Figure 3.7 Block Diagram of Least Squares Positioning Algorithm

$$H = \begin{bmatrix} \frac{v_{nomx} - x_{rcvr1}}{pr_{rcvr1} - v_{nom4}} & \frac{v_{nomy} - y_{rcvr1}}{pr_{rcvr1} - v_{nom4}} & \frac{z_{nomy} - z_{rcvr1}}{pr_{rcvr1} - v_{nom4}} & 1 \\ \frac{v_{nomx} - x_{rcvr2}}{pr_{rcvr2} - v_{nom4}} & \frac{v_{nomy} - y_{rcvr2}}{pr_{rcvr2} - v_{nom4}} & \frac{z_{nomy} - z_{rcvr2}}{pr_{rcvr2} - v_{nom4}} & 1 \\ \frac{v_{nomx} - x_{rcvr3}}{pr_{rcvr3} - v_{nom4}} & \frac{v_{nomy} - y_{rcvr3}}{pr_{rcvr3} - v_{nom4}} & \frac{z_{nomy} - z_{rcvr3}}{pr_{rcvr3} - v_{nom4}} & 1 \\ \frac{v_{nomx} - x_{rcvr4}}{pr_{rcvr4} - v_{nom4}} & \frac{v_{nomy} - y_{rcvr4}}{pr_{rcvr4} - v_{nom4}} & \frac{z_{nomy} - z_{rcvr4}}{pr_{rcvr4} - v_{nom4}} & 1 \end{bmatrix} \quad (3.5)$$

The rest of the algorithm is versatile enough that the H matrix can be expanded to any number of rows to match the number of receivers that are visible to a transmitter at any given time. Equation 3.6 shows an example for a five receiver case.

$$H = \begin{bmatrix} \frac{v_{nomx} - x_{rcvr1}}{pr_{rcvr1} - v_{nom4}} & \frac{v_{nomx} - y_{rcvr1}}{pr_{rcvr1} - v_{nom4}} & \frac{v_{nomx} - z_{rcvr1}}{pr_{rcvr1} - v_{nom4}} & 1 \\ \frac{v_{nomx} - x_{rcvr2}}{pr_{rcvr2} - v_{nom4}} & \frac{v_{nomx} - y_{rcvr2}}{pr_{rcvr2} - v_{nom4}} & \frac{v_{nomx} - z_{rcvr2}}{pr_{rcvr2} - v_{nom4}} & 1 \\ \frac{v_{nomx} - x_{rcvr3}}{pr_{rcvr3} - v_{nom4}} & \frac{v_{nomx} - y_{rcvr3}}{pr_{rcvr3} - v_{nom4}} & \frac{v_{nomx} - z_{rcvr3}}{pr_{rcvr3} - v_{nom4}} & 1 \\ \frac{v_{nomx} - x_{rcvr4}}{pr_{rcvr4} - v_{nom4}} & \frac{v_{nomx} - y_{rcvr4}}{pr_{rcvr4} - v_{nom4}} & \frac{v_{nomx} - z_{rcvr4}}{pr_{rcvr4} - v_{nom4}} & 1 \\ \frac{v_{nomx} - x_{rcvr5}}{pr_{rcvr5} - v_{nom4}} & \frac{v_{nomx} - y_{rcvr5}}{pr_{rcvr5} - v_{nom4}} & \frac{v_{nomx} - z_{rcvr5}}{pr_{rcvr5} - v_{nom4}} & 1 \end{bmatrix} \quad (3.6)$$

Once the H matrix is calculated, it is necessary to calculate the phase range from the nominal position chosen to each of the receivers. This term will be represented by $phrc_i$ where the i represents the ith receiver and is given in Equation 3.7. This value is

$$phrc_i = \sqrt{\left((v_{nomx} - x_{rcvri})^2 + (v_{nomy} - y_{rcvri})^2 + (v_{nomz} - z_{rcvri})^2\right)} + v_{nom4} \quad (3.7)$$

necessary for calculating the *differenced* phase range measurement, which is in turn used to find a delta term to be added to the nominal position. This differenced phase range measurement is the difference in the observed phase range, given in Equation 2.18, and the calculated phase range of Equation 3.7. It will be a vector with the number of rows equal

to the number of receivers in view. It is then necessary to calculate the delta to add to the nominal position. This delta is given in Equation 3.8, where $\overline{\text{diffpr}}$ is a vector of measured phase ranges subtracted from the calculated phase ranges.

$$\overline{\Delta v_{nom}} = (H^T H)^{-1} H^T \overline{\text{diffpr}} \tag{3.8}$$

The $\overline{\Delta v_{nom}}$ term is then added to the nominal position vector $\overline{v_{nom}}$ to determine a new nominal position. This procedure is then repeated until the delta becomes small enough for a criterion set by the user, which can be the number of loops or a minimal change in the nominal position.

3.3.2 Attitude Determination. Once the positions of all the transmitters have been found, solutions of attitude (heading, pitch, and roll) for the aircraft can be found. Since there will be some pitch and roll from the pod contributing to a false calculation, these need to be estimated as well. Therefore, the initial set of data, at time t_0 will not be used to calculate position or attitude. Instead, the data from time t_o will be used to aid in the calculations at the following sampling time, which will be referred to as time t_1.

At time t_1, it is possible to calculate an approximate speed of the aircraft between t_0 and t_1. Since the system is assumed to be running on one second updates, the approximate speed, in feet per second, will be a simple distance calculation, shown in Equation (3.9), using transmitter one's location at t_0 and at t_1. Utilizing this approximate

$$\text{speed} = \sqrt{\left((x_{xmtr1}(t_1) - x_{xmtr1}(t_0))^2 + (y_{xmtr1}(t_1) - y_{xmtr1}(t_0))^2 + (z_{xmtr1}(t_1) - z_{xmtr1}(t_0))^2\right)} \tag{3.9}$$

speed, the approximate roll and pitch of the pod can be calculated in the same manner as mentioned in Section 3.2.2, utilizing an average altitude of the transmitters. Once these estimations of pod pitch and roll have been calculated, the attitude for the aircraft can be estimated.

The first aspect of attitude to be determined is the pitch of the aircraft. To calculate the pitch involves transmitters one and two or transmitters three and four. The transmitters must be used in these sets as one and three or one and four would not work properly. Equation 3.10 provides the relation used to determine pitch. Recall that the z-coordinates are in the latitude, longitude, altitude frame. The nine in the formula represents the known separation of

$$\text{aircraft pitch} = \sin^{-1}((z_{xmtr1} - z_{xmtr2}) / 9) - (\text{est pod pitch})\cos(\phi - \text{est pod roll}) \quad (3.10)$$

transmitters one and two on the pod. Note that the estimated pod pitch, multiplied by a factor of the roll, is subtracted from the pitch calculated. This is to remove the effects of the pod's pitch from the calculated pitch to determine the aircraft's actual pitch. Also, since there are two sets of transmitters, it would be possible to calculate a pitch based on transmitters three and four and average the results.

The roll of the aircraft can be calculated utilizing an altitude difference in transmitters one and three or transmitters two and four. Due to the fact that there may be pitch of the aircraft, it is again necessary to keep these combinations in this order. Since the primary effects of the wing flexure are symmetric as discussed in Section 2.3, the

imaginary line connecting transmitters one and three will be parallel to the roll of the aircraft (see Figure 3.8). Therefore, it is possible to calculate this roll utilizing Equation (3.11). In Equation (3.11), the distance between transmitters one and three is calculated

$$\text{aircraft roll} = \sin^{-1}((z_{xmtr1} - z_{xmtr3}) / ((\text{distance transmitter 1 to 3}) \cos(\theta)) \qquad (3.11)$$

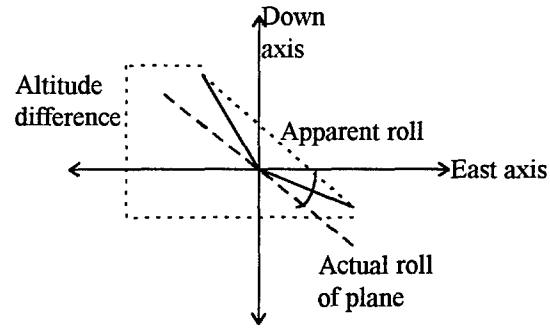

Figure 3.8 Parallel Nature of Actual Roll to Apparent Roll

using the standard distance formula of Equation (3.4). This is to account for the fact that wing flexure makes the apparent distance less than the thirty-one feet separation in the rigid wing case. Examining Figure 3.8 should clarify this relationship. Again, since there are two sets of transmitters it would be possible to calculate the estimated roll for the other set and average the estimated rolls.

Since the aircraft roll and pitch are necessary for calculating each other, an iterative technique was implemented. The correction factor for the pitch is a very small number (-0.5 degrees or less) due to the flight conditions considered in this research. Therefore, the pitch was calculated assuming no roll effects on this term and roll was calculated using this pitch. The program then went back to the pitch equation and used

the new roll to calculate the value of pitch and then the roll based on that pitch. This was repeated until the changes in roll and pitch values were less than 0.0001.

To calculate the heading of the aircraft, the first step is to find a calculated heading based upon the estimated navigation frame coordinates of the transmitters. To do this, take the x, y, z geocentric coordinates found by the least squares algorithm and convert these to navigation frame coordinates using the transpose of the matrix given in Equation (3.3). The calculated heading is then equal to:

$$\text{calculated heading} = \tan^{-1}\left(\frac{\text{ynav}_{\text{xmtr1}} - \text{ynav}_{\text{xmtr2}}}{\text{xnav}_{\text{xmtr1}} - \text{xnav}_{\text{xmtr2}}}\right) \tag{3.12}$$

Now it is necessary to go through a derivation of the locations of transmitter one and transmitter two. Equation (3.13) shows the locations of transmitters one and two in

$$\text{xmtr}_1^b = \begin{bmatrix} A \\ W+D \\ B+C \end{bmatrix} \qquad \text{xmtr}_2^b = \begin{bmatrix} -A \\ W-D \\ B-C \end{bmatrix} \tag{3.13}$$

body-frame coordinates. Note that the A represents the term 4.5 cos(pod pitch), W represents the 15.5 cos(pod roll) term, D represents the -4.5 sin(pod roll)sin(pod pitch) term, B represents the -15.5 sin(pod roll), and C represents the -4.5 sin(pod pitch term)cos(pod roll) term. Multiplying these by the body to navigation frame direction cosine matrix yields navigation frame coordinates of the transmitters (shown in Equation (3.14)). Note that the c represents the cosine of an angle and s

represents the sine of an angle and A, W, B, and C are as defined in Equation (3.13). The heading, pitch, and roll are represented as in Equation (3.2).

$$\text{xmtr}_1^n = \begin{bmatrix} A\,c\theta\ c\psi + (W+D)(s\phi\, s\theta\, c\psi - c\phi\, s\psi) + (B+C)(c\phi\, c\psi\, s\theta + s\phi\, s\psi) \\ Ac\psi + (W+D)(s\phi\, s\theta\ s\psi + c\phi\, c\psi) + (B+C)(c\phi\, s\theta\ s\psi - s\phi\, c\psi) \\ -As\theta + (W+D)c\theta\, s\phi\ + (B+C)c\theta\ c\phi \end{bmatrix}$$

$$\text{xmtr}_2^n = \begin{bmatrix} -A\,c\theta\ c\psi + (W-D)(s\phi\, s\theta\, c\psi - c\phi\, s\psi) + (B-C)(c\phi\, c\psi\, s\theta + s\phi\, s\psi) \\ -Ac\psi + (W-D)(s\phi\, s\theta\ s\psi + c\phi\, c\psi) + (B-C)(c\phi\, s\theta\ s\psi - s\phi\, c\psi) \\ As\theta + (W-D)c\theta\, s\phi\ + (B-C)c\theta\ c\phi \end{bmatrix} \quad (3.14)$$

In this case, the calculated heading from Equation (3.12) would equal the expression shown in Equation (3.15), with all variables as in Equation (3.14).

$$\text{calc heading} = \tan^{-1}\left(\frac{Ac\theta\, s\psi + D(s\phi\, s\theta\, s\psi + c\phi\, c\psi) + C(+c\phi\, s\theta\, s\psi - s\phi\, c\psi)}{Ac\theta\, c\psi + D(s\phi\, s\theta\, c\psi - c\phi\, s\psi) + C(s\phi\, s\psi + c\phi\, c\psi\, s\theta)}\right) \quad (3.15)$$

Given that a heading can be calculated from the estimated navigation frame coordinates, and that pitch and roll are already known, Equation (3.16) can be expressed as:

$$\text{calculated heading} = \tan^{-1}\left(\frac{P_1 s\psi - P_2 c\psi + P_3 s\psi + P_4 s\psi + P_5 c\psi}{P_1 c\psi + P_2 s\psi + P_3 c\psi + P_4 c\psi - P_5 s\psi}\right) \quad (3.16)$$

where P_1 equals $A\cos\theta$, P_2 equals $C\sin\phi$, P_3 equals $C\sin\theta\cos\phi$, P_4 equals $D\sin\theta\sin\phi$, and P_5 equals $D\cos\phi$. This can be regrouped as:

$$\text{calculated heading} = \tan^{-1}\left(\frac{\beta_1 s\psi - \beta_2 c\psi}{\beta_2 s\psi + \beta_1 c\psi}\right) \quad (3.17)$$

where β_1 equals $P_1+P_3+P_4$ and β_2 equals P_2-P_5. Since it is necessary to isolate the actual heading for the solution, multiplying the top and bottom of the fractional expression by $1/\cos\psi$ yields:

$$\text{calculated heading} = \tan^{-1}\left(\frac{\beta_1 \tan\psi - \beta_2}{\beta_2 \tan\psi + \beta_1}\right) \tag{3.18}$$

If the tangent of each side is taken, and then simple regrouping is done:

$$\tan\psi(\beta_2 \tan(\text{calculated heading}) - \beta_1 = -\beta_1 \tan(\text{calculated heading}) - \beta_2 \tag{3.19}$$

Dividing through and taking the arc tangent yields the expression for aircraft heading shown in Equation (3.20).

$$\psi = \tan^{-1}\left(\frac{-\beta_1 \tan(\text{calculated heading}) - \beta_2}{\beta_2 \tan(\text{calculated heading}) - \beta_1}\right) \tag{3.20}$$

3.3.3 Aircraft Position. The last step is to determine the position of the aircraft. This can be done by a simple averaging of coordinates of the four transmitters. However, if this is done, the result will actually be the position of the point that is the midpoint of the imaginary line connecting transmitters in Figure 3.9. In order to compensate for this difference, an expression of the distance between the midpoint and the origin of the figure was derived and is shown in Equation 3.21. This formula assumes the flexure of the wing to be almost linear; allowable considering the maximum roll angle of the pod.

$$\text{adjustment} = .5 * \text{length(xmtr1 to xmtr3)} * \tan(\text{estimated pod roll}) \tag{3.21}$$

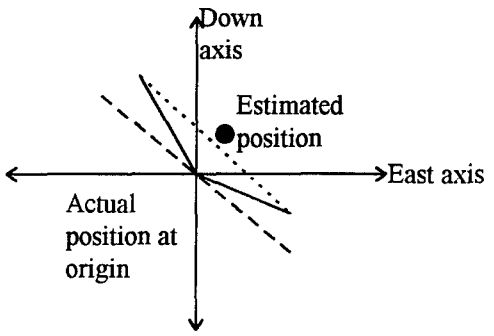

Figure 3.9 Error in Aircraft Position Calculation

Due to the symmetric nature, all the coordinates in the body frame average out, except the z coordinate due solely to pod roll that is common in all the transmitter locations. Hence this offset is put in as the z-body coordinate and converted to a navigation frame vector by the direction cosine matrix of Equation 3.4. These navigation frame coordinates are then added to the averaged solution to find the corrected solution (corrected for pod pitch and roll) for the location of the plane at the time under consideration. This solution is then used as the initial position guess, $\overline{v_{nom}}$, in the next iteration until the simulation is complete.

3.4 van Graas' Method. This section is a brief overview of a method Dr. van Graas is utilizing in his systems to determine attitude (23). He uses a standard CPGPS system with four receivers on the aircraft (one on each wing, one on the nose, and one on the tail) and four or more satellite transmitters. The navigation frame coordinates of the receivers are calculated as the plane is flying. Since the body frame coordinates of the

receivers are known beforehand, he computes the direction cosine matrix, of the same form as Equation (3.2) as

$$C_b^n = R_s R_a^T (R_a R_a^T)^{-1} \tag{3.21}$$

where R_S is a 3x4 matrix containing the navigation-frame coordinates of the receivers and R_a is a 3x4 matrix containing the body-frame coordinates of the receivers. From this computed direction cosine matrix, he is able to calculate heading, pitch, and roll.

3.4 Summary. This chapter explained the simulation models to be used in generating the results. The first part discussed the reference model and then gave a description of the proposed attitude and position estimation method. Also mentioned was the technique employed by Dr. van Graas to estimate attitude. Chapter IV will discuss the results of simulation runs implementing these models and estimation methods.

IV. Results

4.1 Overview

This chapter will discuss the specifications of the simulation runs. It will then offer a brief description of the results for the reader. Complete plots of the results are supplied in Appendices B and C.

4.2 Simulation Specifications.

This section describes the specifics of the two flight profiles used to derive results from as well as describing which routines were utilized. Please note that in all cases, fifteen runs were used for Monte Carlo analysis.

4.2.1 Flight Profile 1. For flight profile 1, the following flight path was used: straight and level flight for the first ten seconds, followed by a pitch maneuver of -2.5 degrees per second for eight seconds and a pitch maneuver of +2.5 degrees per second for eight seconds, then finishing with four seconds of straight and level flight. Throughout the flight path, the aircraft maintained 600 feet per second velocity with constant heading of 0°. The flight path began at 34.75° north, 84.5^0 west, and 10000 feet above sea level. This flight plan allowed the ability to easily check results and verify that all routines were functioning properly.

Results for this plan were run considering a four receiver case and a five receiver case. The four-receiver case employed receivers that were effectively laid out in a box-like formation encompassing the flight path of the aircraft from beginning to end. To accomplish this all the receivers were placed at 1000 feet above sea level and with receiver

one positioned at 34.5° north latitude, 84° west longitude, receiver two at 34.5° north, 84.5° west, receiver three at 35° north, 84° west, and receiver four at 35° north, 84.5° west. For the five-receiver case, a fifth receiver was placed at an elevation 7000 feet higher than the other receivers at 34.55° north, 84.05° west. Also, as a verification method, transmitter position estimates that were assumed to be within ten centimeters of true positions were tested with the algorithms developed. To generate those accuracies, the actual transmitter locations were used and had a zero mean Gaussian random number with covariance of (2/12) ft^2 added to represent the error involved.

4.2.2 Flight Profile 2. This flight profile's attitude and speed are identical to Flight Profile 1 through the first ten seconds. However, the flight path started at 34.6° north instead of 34.75° north. At the eleven second point, the aircraft undergoes a heading maneuver of three degrees per second, a pitch maneuver of minus five degrees per second and a rolling maneuver of five degrees per second. The heading and pitch rates remain constant for eight seconds and then are reversed to +5° per second pitch and -3° per second heading . The aircraft rolls at five degrees per second for four seconds, reverses to minus five degrees per second for eight seconds, and reverses one last time to five degrees per second for four seconds to get back to straight and level flight. This plan allows for the check of the routines under more realistic conditions of aircraft flight.

The results for this flight plan were run only for the five-receiver case. This decision was made due to the results of the four-receiver case described in Section 4.2.1 and described in Section 4.3.1. Once again the routine was run to generate ten centimeter accuracies and those results were then tested in the algorithms. As an extra test for this

flight path, Dr. van Graas' method, described in Section 3.4, was also attempted and examined.

4.3 Results of Flight Profile 1.

4.3.1 Four-Receiver Case. The initially considered scenario was the four-receiver case. In this situation, all the receivers were laid out in an environment much like a person would expect to find in the midwest with receivers at relatively the same altitudes. A key element in GPS accuracy is the geometric dilution of precision (GDOP). It is calculated by Equation (4.1)

$$\text{GDOP} = \sqrt{\left(\text{trace}\,(H^T H)^{-1}\right)} \tag{4.1}$$

where H is as defined in Equation (3.5). Overall GPS positioning errors are equal to the GDOP multiplied by the user ranging errors (clock biases, drifts, atmospheric delays). Therefore, high GDOP causes high positioning errors.

As one would expect given the receiver locations, GDOP was incredibly poor with the best GDOP for the flight being sixty-five and ranging up to one hundred thirty-three. Figure 4.1 shows the GDOP versus time for the length of the flight.

Since the GDOP was so poor, it was virtually impossible to get good estimates of the transmitter locations and the aircraft's location and attitude. Figure 4.2 shows the transmitter one position error along with the aircraft position error. The transmitter position error varies from fifty feet to one hundred twenty five feet, with the standard deviation being around fifty feet for the entire flight. Interestingly, due to the averaging of

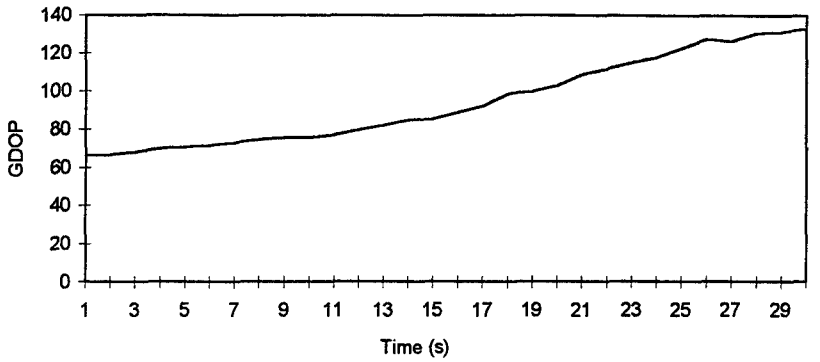

Figure 4.1 GDOP versus Time - Flight Profile 1
Four-Receiver Case

transmitter locations, the aircraft position error only varied from twenty to sixty feet with the standard deviation being around twenty-five feet the entire time.

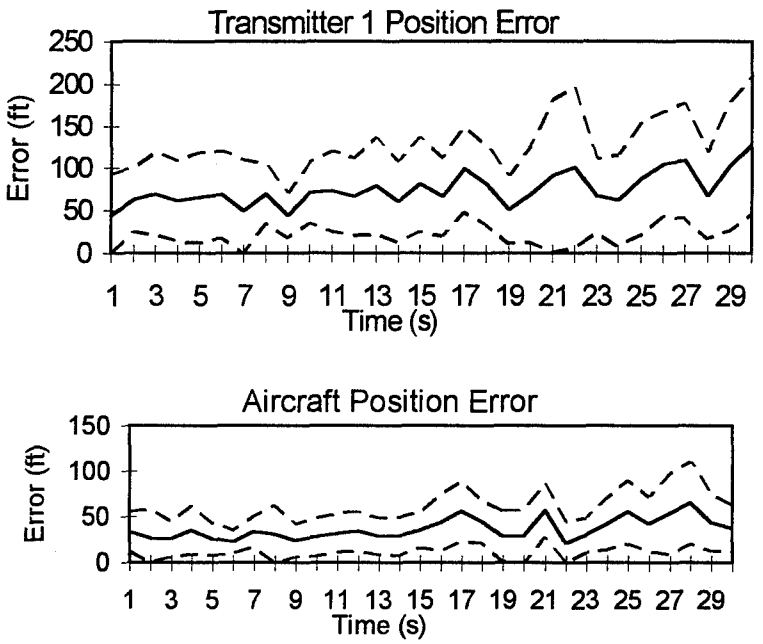

Figure 4.2 Transmitter One and Aircraft Position Errors
Flight Profile 1, Four-Receiver Case

Figure 4.3 shows the attitude errors for the aircraft in this trial. As a result of transmitter position estimates that were highly inaccurate, the attitude determination

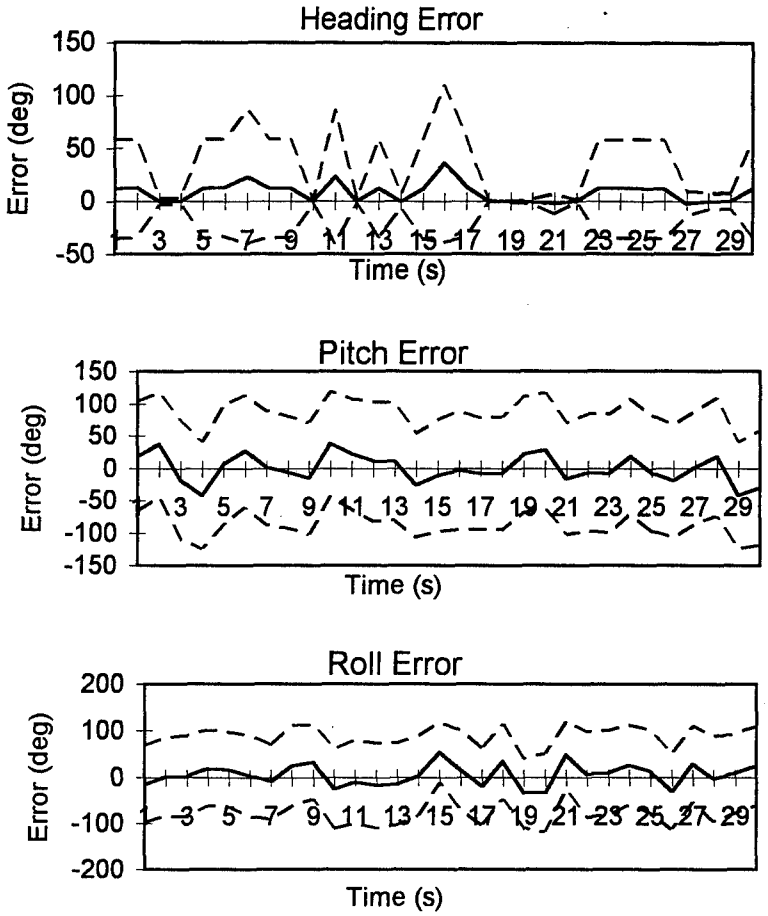

Figure 4.3 Attitude Errors
Flight Profile 1, Four-Receiver Case

algorithms worked poorly for this test. It is clear that the heading, pitch, and roll estimates were essentially worthless, due to the large errors in terms used to calculate those angles.

4.3.2 Five-Receiver Case. For this case, the fifth receiver that was added at an elevation significantly higher than the other four (7000 feet) significantly impacted the GDOP. Instead of varying in the sixties through hundreds, the GDOP ranged in the teens, still unusable for the accuracies desired. Figure 4.4 shows the GDOP versus time for this case.

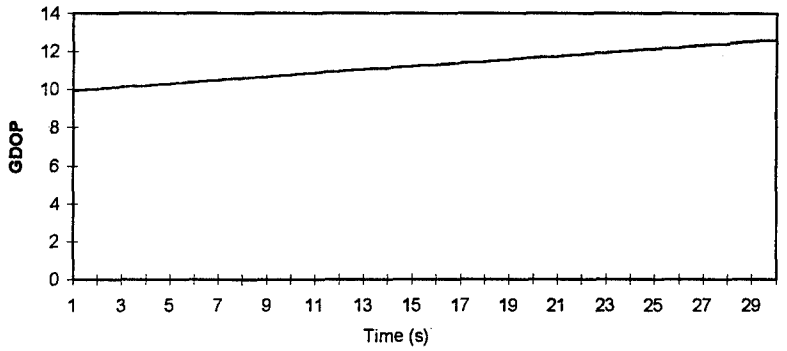

Figure 4.4 GDOP versus Time - Flight Profile 1
Five-Receiver Case

Since the GDOP was significantly better than before, the transmitter position errors were also significantly improved, as well as the overall aircraft position error. Figure 4.5 illustrates transmitter one position error and aircraft position error for this trial.

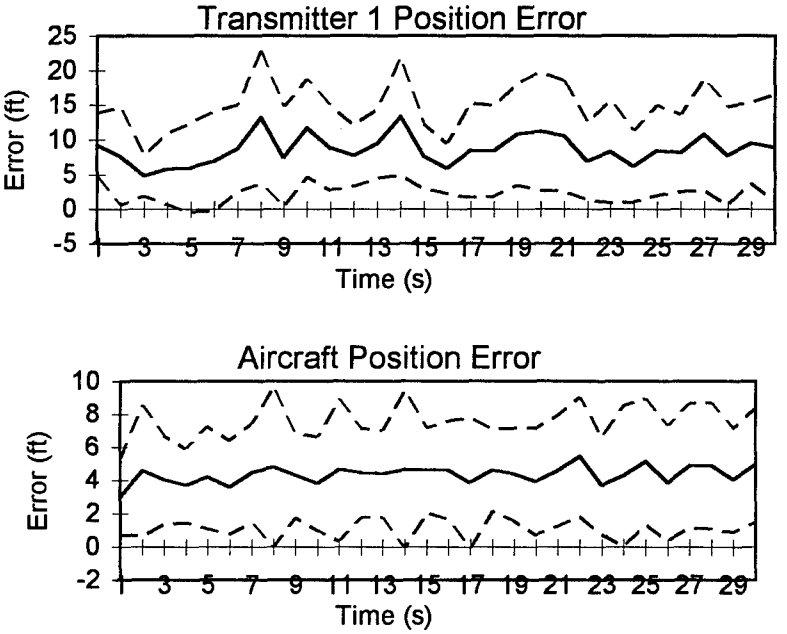

Figure 4.5 Transmitter One and Aircraft Position Errors
Flight Profile 1, Five-Receiver Case

In this case, the average transmitter position error is around ten feet with about a five foot standard deviation and the aircraft position error is around four feet with a three foot standard deviation. Again, the improvement in aircraft position error is due to the averaging of the four transmitter locations.

Figure 4.6 shows the attitude errors. Again, the results are very poor due to the errors in the transmitter position estimates, although the standard deviations were significantly reduced relative to the four receiver case.

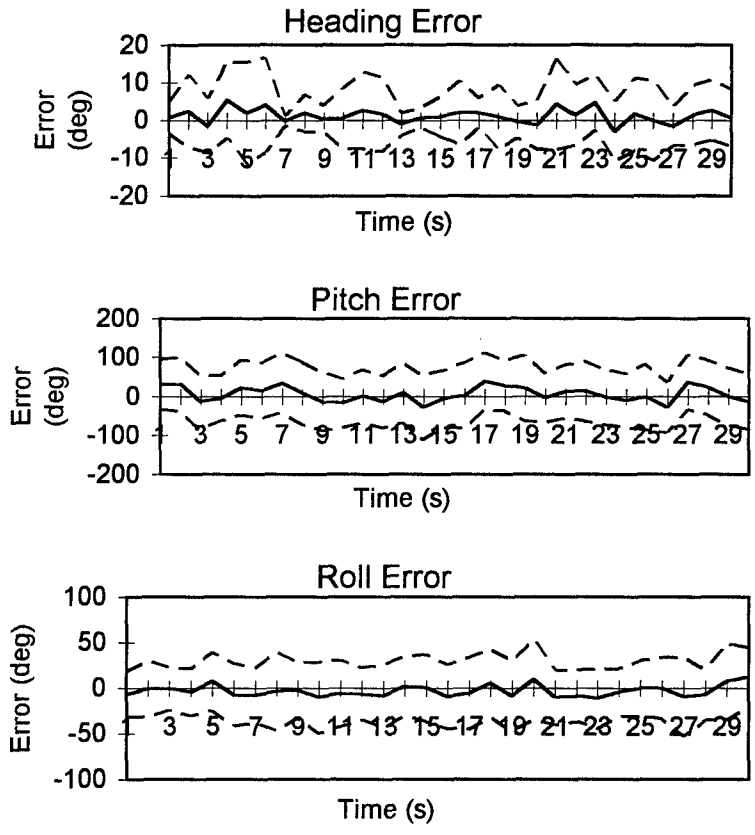

Figure 4.6 Attitude Errors
Flight Profile 1, Five-Receiver Case

4.3.3 Assumed Transmitter Location Accuracy. This trial considered transmitter position estimation errors that were zero mean with a standard deviation of ten centimeters or about four inches. When considering accuracies that were that good, the algorithms worked significantly better. Figure 4.7 shows the aircraft position error for the

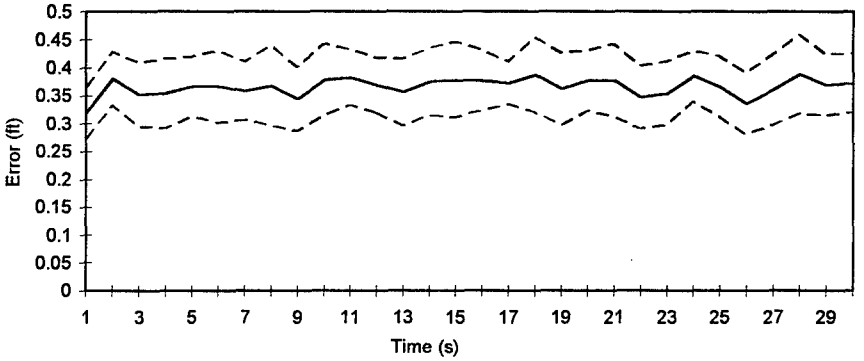

Figure 4.7 Aircraft Position Error
Flight Profile 1, Accurate Estimates

flight. The position error is typically around 0.36 feet with a standard deviation of approximately 0.06 feet. These results correspond to about an eleven centimeter miss, near the target of ten centimeters provided by Holloman (15).

Perhaps the most significant impact came in the attitude determination. Figure 4.8 shows the attitude errors for the flight. The pitch error was the worst with a zero mean and a 0.8 degree standard deviation. Heading had good accuracies with a zero mean and standard deviation of 0.8 degrees. Roll had excellent results showing zero mean with a 0.3 degree standard deviation. These results are more in line with what was expected and desired for this experiment.

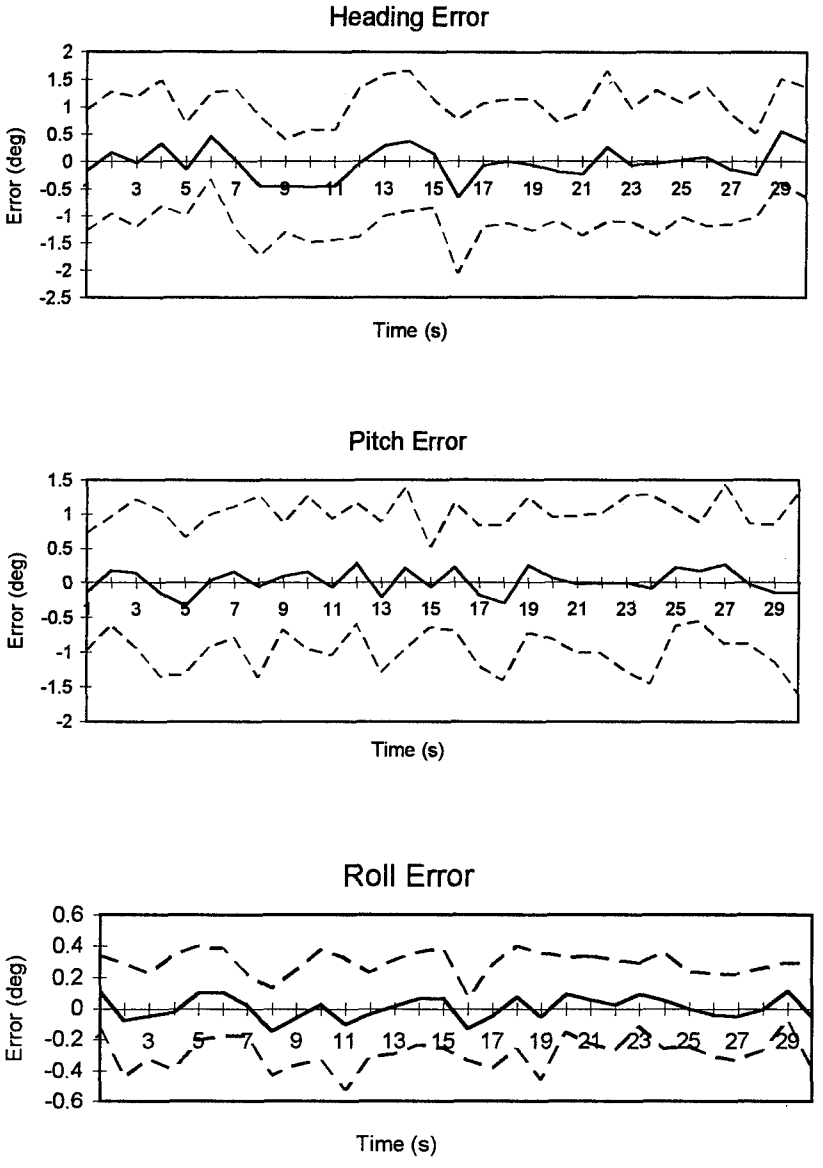

Figure 4.8 Attitude Errors
Flight Profile1, Accurate Estimates

4.4 Results of Flight Profile 2.

4.4.1 Five-Receiver Case. For flight profile two, the results of the four receiver case could have been easily projected beforehand due to the GDOP problems, so only the

five receiver case was considered. Figure 4.9 shows the GDOP for this flight. It is a little poorer due to the heading changes taking the aircraft out of the path initially used, although still significantly better than for the four receivers of flight one.

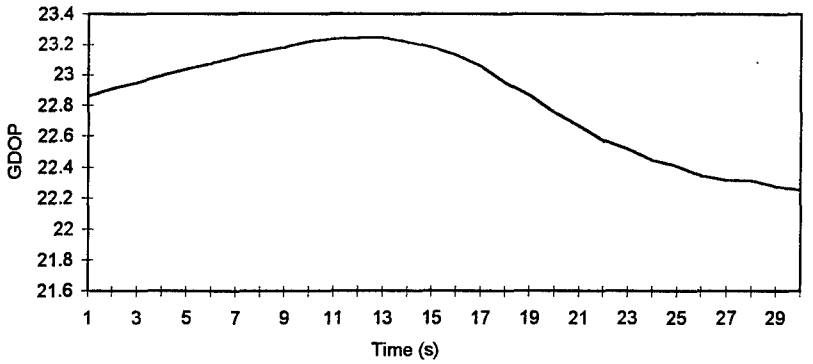

Figure 4.9 GDOP versus Time - Flight Profile 2
Five Receiver Case

Figure 4.10 shows these errors for transmitter one and the aircraft. Due to the

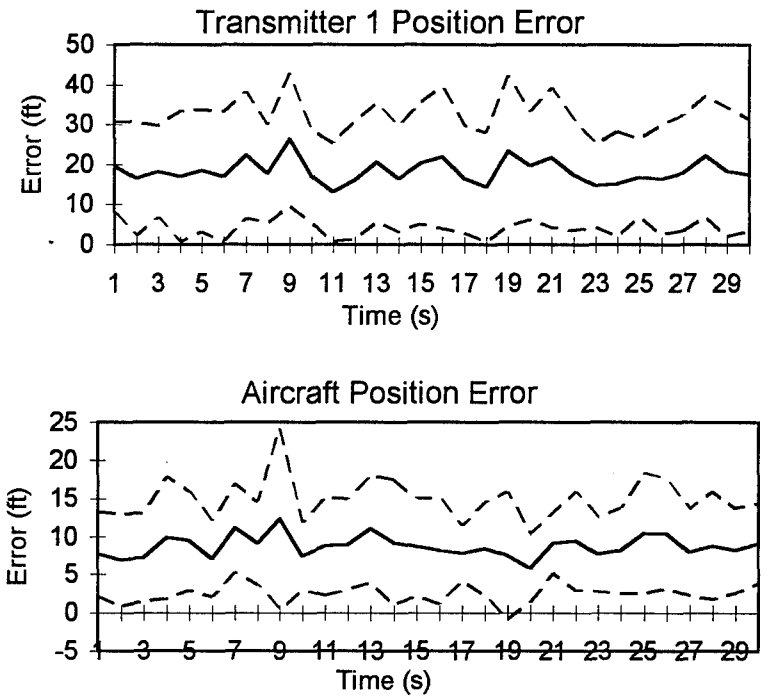

Figure 4.10 Transmitter One and Aircraft Position Errors
Flight Profile 2, Five-Receiver Case

GDOP being approximately doubled, the position errors of the transmitters and the aircraft are about doubled from the previous results, although still good compared to the four receiver case of flight one. Once again, these results are generally unacceptable for the desired accuracy previously noted.

As in the previous five receiver case, the large positioning errors led to huge errors in the attitude estimates. Figure 4.11 shows the heading, pitch, and roll errors for this

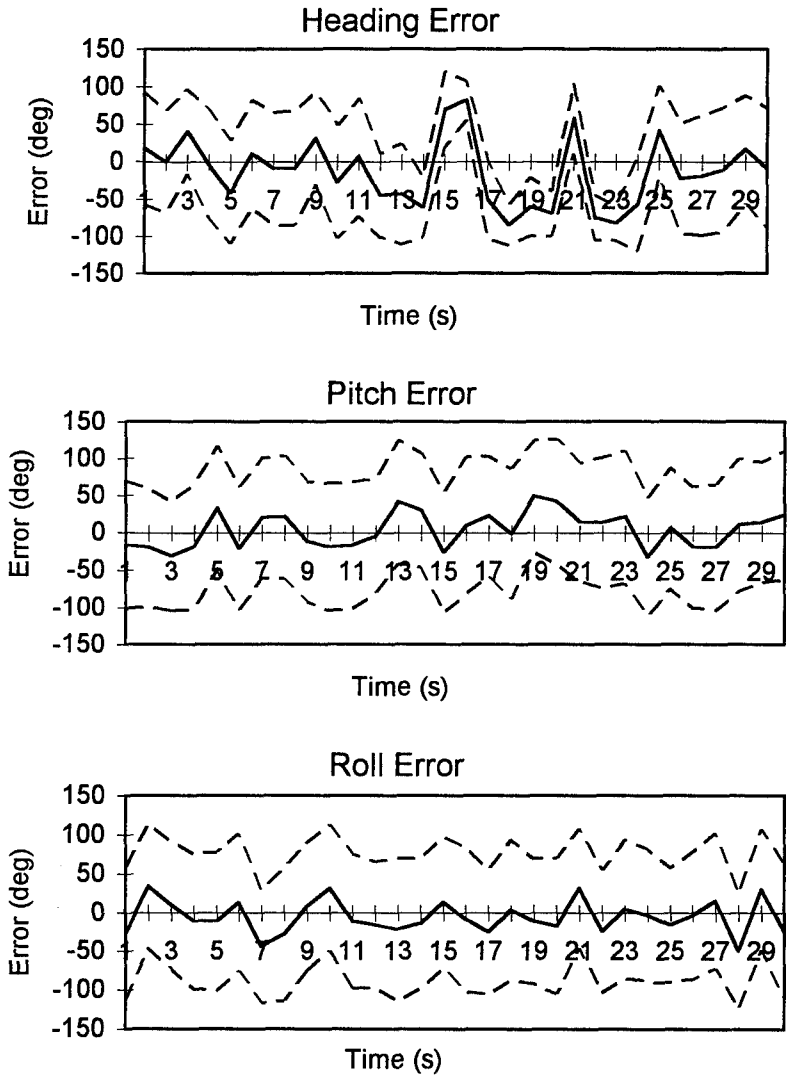

Figure 4.11 Attitude Errors
Flight Profile 2, Five-Receiver Case

case. The errors were essentially zero mean but only because of the huge swings from positive to negative errors. Also note that while the errors may be close to zero mean, the standard deviations are again large, up to fifty to one hundred degrees in some cases.

4.4.2 Assumed Transmitter Location Accuracy. Again the flight profile was run assuming the transmitter position estimates are within the desired accuracy. Figure 4.12

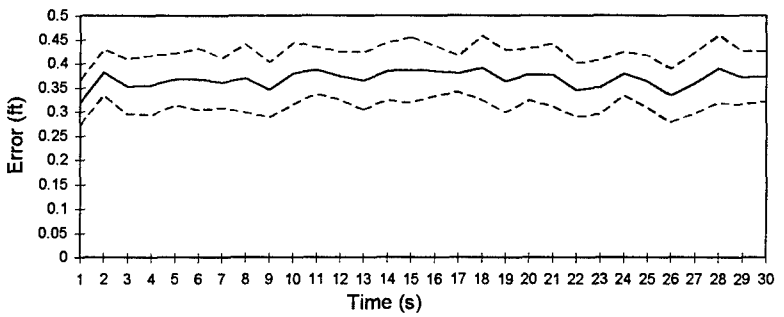

Figure 4.12 Aircraft Position Error
Flight Profile 2, Accurate Estimates

shows the aircraft position error. For this case the aircraft position error was again around 0.36 feet with the standard deviation around 0.06 feet meeting the desired goal for accuracy.

Figure 4.13 illustrates the attitude parameter errors for this case. Again, pitch has the worst results with zero mean and standard deviation of one degree. Heading and roll exhibited similar behavior to previous results with zero mean errors and standard deviations of 0.6 and 0.2 degrees respectively.

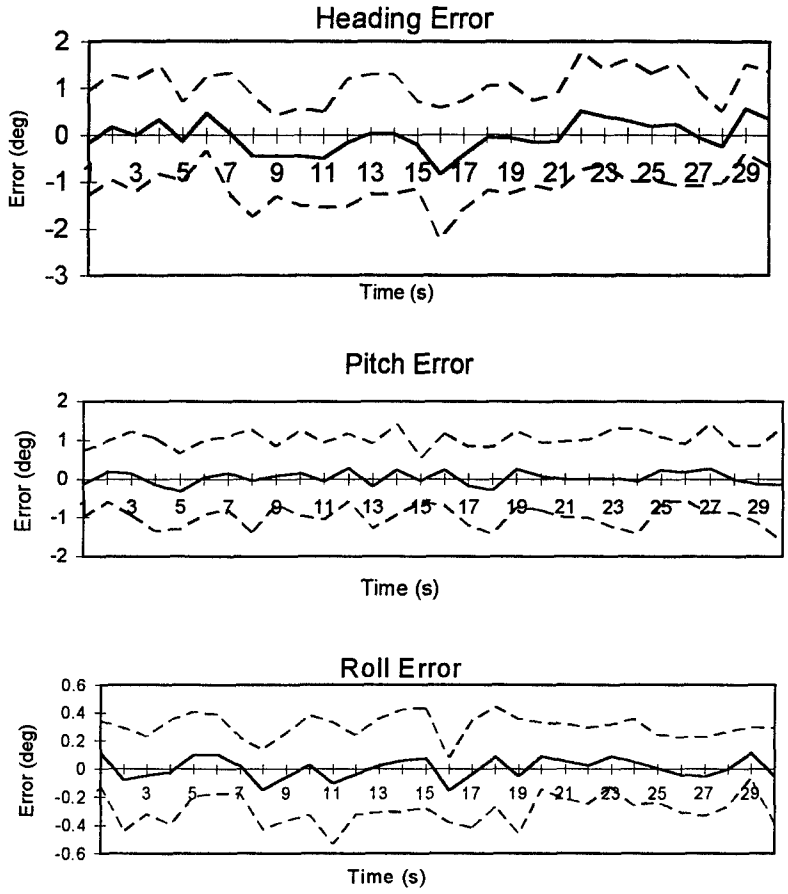

Figure 4.13 Attitude Errors
Flight Profile 2, Accurate Estimates

4.4.3 van Graas' Method. For this flight plan, a decision was made to utilize Dr. van Graas' algorithm briefly described in Section 3.4. When this algorithm was run, it ended up generating poor results for the attitude parameters (see Appendix C for plots). This was due to the problems Matlab had in taking the $R_s R_a^T (R_a R_a^T)^{-1}$ term. The key reason for this was that the transmitters were approximately in the same plane so none of them had significantly larger body frame z-coordinates than the others making the $(R_a R_a^T)^{-1}$ matrix ill-conditioned. In van Graas' test, one receiver was on the tail of the aircraft giving it a large altitude variation in the body frame.

4.5 Summary. This chapter covered three topics:

1. It outlined the two flight test profiles that were used to check the position and attitude determination algorithms and the van Graas method.

2. It presented the results generated by the test profiles using both an algorithm to determine transmitter location estimates and the theoretically achievable transmitter location estimates.

3. It provided a discussion of why some results were poor and why Dr. van Graas' algorithm did not work correctly in this situation.

Chapter V, Conclusions, will summarize what the reader can interpret from the tests.

V. Conclusions

5.1 Overview

This chapter will summarize the results of this thesis and propose several conclusions that can be drawn from the results. Also, it will provide some recommendations for areas of further research.

5.2 Conclusions.

The tests run in this thesis generate several conclusions that are readily apparent. First, the four-receiver case where all receivers are at about the same altitude is not usable in any way for this system. The GDOP created is simply too poor for use, although it could be enhanced by moving the receivers closer together. For the five-receiver case, the GDOP did improve, but was still not good enough for practical use with this system. Aircraft positioning in both these cases was good, considering the poor transmitter location estimates, due to the averaging of transmitter locations, but it was still not good enough for use by CIGTF, again due to poor GDOP. This infers that this system will be most viable in mountainous regions where great altitude differences are readily available.

When the algorithms were run utilizing assumed transmitter position estimate accuracies of ten centimeters, the algorithm functioned extremely well, demonstrating the validity of these algorithms for future use, assuming that the problems due to poor GDOP can be fixed.

Finally, Dr. van Graas' algorithm, while extremely capable for the correct geometry as he utilized, is inefficient in this situation. As discussed before, this is

primarily due to the transmitters being in the same plane in the body frame system, not any fault with the algorithm itself.

5.3 Recommendations for Future Research.

1. Research must be done to develop a method of generating the ten centimeter accuracy in the transmitter location estimates. Without this level of accuracy, none of the other variables can be estimated to within a reasonable error. The biggest key to achieving this level of accuracy is to improve the GDOP to four or less. This may be achieved by increasing the number of receivers available.

2. It would be useful to examine the possibilities of filtering on the solutions generated. This could include checking for any large swings in the heading, pitch, or roll greater than the rates the aircraft would go through, or to develop a maximum likelihood estimator for transmitter locations.

3. Another good experiment would be to attempt to model more factors in the reference model. These could include the clock drift term for the satellite and receiver clocks and the frequency components of the wing bending that are currently modeled by simply increasing the CPGPS measurement noise strength.

4. Perhaps the best alternative would be to change the configuration to utilize transmitters located much like van Graas' system and thus be able to utilize his matrix multiplication method while focusing research energies on improving GDOP for this situation.

5.4 Summary.

In summary, this research investigated one potential method of attitude and position determination for an aircraft utilizing an inverted CPGPS system. This method worked well when individual transmitter position accuracies were good enough to yield meaningful results and has potential for implementation as a computationally quick system.

Appendix A. Matlab Programs

This appendix contains listings of the Matlab programs written for this research. The first group of programs were used to generate the reference model, while the second set were the aircraft attitude and position algorithms. Please note that each new program begins with a line '****** <Filename> ******' to help the reader distinguish these. Also, each program is given a separate page, while functions written were placed together when possible.

```
% ****** FLIGHT.M ******
%
% This program simulates an aircraft in flight, compensating for
% the wing flexure in order to get accurate simulations of the carrier-
% phase measurements that would be received in actual application.
% It uses several subroutines to perform this:
% LOCPOD.M - finds the location of the pod based upon
%                aircraft's heading and attitude
% CPGPSSIM.M - simulates the carrier-phase measurement
% SETUP.M - sets the initial values for flight and for receiver location
% READINFO.M - reads in next times information - speed, roll, pitch, yaw, etc.
% CHANGELOC.M - moves the plane from current time to next time

% call to setup to get initial values established
numtrials=15;
for trial=1:numtrials
        dummy=trial
        setup;
        locpod;
        cpgpssim;

        for time=1:endsim
                heading=flightmatrix(time,1);
                pitch=flightmatrix(time,2);
                roll=flightmatrix(time,3);
                spdplane=flightmatrix(time,4);
                dummy=time;
                changloc;
                locpod;
                cpgpssim;
        end
        xmtr1x(trial,:)=xposition(1,:);
        xmtr1y(trial,:)=yposition(1,:);
        xmtr1z(trial,:)=zposition(1,:);

        xmtr2x(trial,:)=xposition(2,:);
        xmtr2y(trial,:)=yposition(2,:);
        xmtr2z(trial,:)=zposition(2,:);

        xmtr3x(trial,:)=xposition(3,:);
        xmtr3y(trial,:)=yposition(3,:);
        xmtr3z(trial,:)=zposition(3,:);

        xmtr4x(trial,:)=xposition(4,:);
        xmtr4y(trial,:)=yposition(4,:);
```

```
        xmtr4z(trial,:)=zposition(4,:);
end
```

```
% ******SETUP.M******
%
% This routine declares the initial variables and any constants necessary
% for the program. Note that with the routines for converting to lat, long,
% and altitude, or for converting other values, the routines are set up to
% do it in either feet or meters depending upon how the variables or constants
% are declared. Be CERTAIN CONSISTENCY is maintained. After each declaration
% a short description of the variable or constant follows.

radiusearth = 20925696;      % radius of the earth in feet; in meters: 6378164.9
pod2podlen = 31;             % distance between pods on F-16, in ft.; in m.: 9.449
podlength = 9;               % distance between pod-mounted transmitters, in ft.;
                             % in m.: 2.743
soundspeed = 1117;           % speed of sound in air (sea level) in ft/s; in m.:340

% Defining coordinates for permanently stationed receivers and the pseudolite
% on the mountain. Given in x,y,z to ease distance calculations.
% NOTE to yourself: you will need to reorganize these in order to allow for
% changing rcvrs in use -- something like x1rcvr, x2rcvr thru # or rcvrs

xxmtr(5) = 1717561.46;       % x coordinate of the mountain pseudolite
yxmtr(5) = -17057118;        % y coordinate of the mountain pseudolite
zxmtr(5) = 12003920.1;       % z coordinate of the mountain pseudolite

xrcvr(1) = 1802722.78;       % x coordinate of receiver 1
yrcvr(1) = -17151761.5;      % y coordinate of receiver 1
zrcvr(1) = 11853011.1;       % z coordinate of receiver 1

xrcvr(2) = 1652978.68;       % x coordinate of receiver 2
yrcvr(2) = -17166839.9;      % y coordinate of receiver 2
zrcvr(2) = 11853011.1;       % z coordinate of receiver 2

xrcvr(3) = 1791842.16;       % x coordinate of receiver 3
yrcvr(3) = -17048239.3;      % y coordinate of receiver 3
zrcvr(3) = 12003059.7;       % z coordinate of receiver 3

xrcvr(4) = 1643001.86;       % x coordinate of receiver 4
yrcvr(4) = -17063226.8;      % y coordinate of receiver 4
zrcvr(4) = 12003059.7;       % z coordinate of receiver 4

xrcvr(5) = 1794629.3;        % x coordinate of receiver 5
yrcvr(5) = -17074757.5;      % y coordinate of receiver 5
zrcvr(5) = 11977127.8;       % z coordinate of receiver 5

numrcvrs = 5;
```

```
for i=1:numrcvrs
        xmtr5rng(i)=sqrt((xxmtr(5)-xrcvr(i))^2+(yxmtr(5)-yrcvr(i))^2+(zxmtr(5)-
zrcvr(i))^2);
end

% Defining values of the errors. Clock biases, without clock drifts will be
% considered. Clock drifts can be ignored for this experiment since the
% duration of the experiment is very short and the oscillators are very stable
% for short times. Tropospheric delay is being considered constant over the
% test range.

for i=1:5
        rcvrclk(i) = 13.9+randn;        % receiver clock bias, in ft.
        xmtrclk(i) = 10+randn;          % transmitter clock bias, in ft.
end;
tropdelay = randn;                      % tropospheric delay, in ft. (typically 0-1 ft)


% Define initial values for changing variables.

time = 0;               % starting time for the simulation
endsim=30;              % time to end the simulation

% These are initially set up for a rigid body, straight and level flight
heading = 0;            % starting heading of the aircraft for this sim.
pitch = 0;              % starting pitch of aircraft for this sim.
roll = 0;                       % starting roll of aircraft for this sim.
podroll = 0;            % starting roll of AIM 9 pod for this sim.
podpitch = 0;           % starting pitch of AIM 9 pod for this sim.

norpos(time+1) = 34.6;                  % north, east down coordinates of the aircraft
eastpos(time+1) = -84.1;                % at initial time. North and East are positive,
downpos(time+1) = radiusearth+10000;    % expressed in degrees and down is
                                        % expressed in ft.

[xplane(time+1), yplane(time+1), zplane(time+1)]=lla2ecef(norpos(time+1),
                                                eastpos(time+1), downpos(time+1));
for i=1:4
        xposition(i,1)=xplane(1);       % give initial "transmitter positions"
        yposition(i,1)=yplane(1);       % needed for CPGPS simulations
        zposition(i,1)=zplane(1);
end

spdplane = 600;                 % initial speed of the aircraft, ft/s
```

```
% Matrices to be used for the finding the pod roll and pitch. It will use the TABLE2
% function. This is a routine that will do the interpolation of the values from the F-16
% wing study. For a description of it, see any MATLAB text

pitchtable(1,:) = [0, 0.1, 4000, 10000, 25000, 50000];
pitchtable(2,:) = [0, -0.1, -0.1, -0.1, -0.1, -0.1];
pitchtable(3,:) = [0.6, -0.75, -0.6, -0.5, -0.25, -0.075];
pitchtable(4,:) = [0.85, -2, -1.7, -1.4, -0.8, -0.2];
pitchtable(5,:) = [0.9, -2.25, -2, -1.65, -0.775, -0.23];
pitchtable(6,:) = [0.95, -4.05, -3.5, -3, -1.4, -0.6];
pitchtable(7,:) = [1.1, -3, -2.55, -2.1, -1, -0.8];
pitchtable(8,:) = [1.2, -3.2, -2.6, -2.05, -1.1, -0.6];
pitchtable(9,:) = [1.4, -2.9, -2.475, -1.95, -1, -0.6];
pitchtable(10,:) = [1.5, -2.8, -2.45, -1.9, -0.95, -0.5];

rolltable(1,:) = [0, 0.1, 4000, 10000, 25000, 50000];
rolltable(2,:) = [0, -0.7, -0.8, -0.8, -0.8, -0.8];
rolltable(3,:) = [0.6, -0.2, 0, 0.2, 0.5, 0.5];
rolltable(4,:) = [0.85, -1.5, -1.1, -0.5, 0.1, 0.8];
rolltable(5,:) = [0.9, -2, -1.3, -0.9, 0.2, 0.9];
rolltable(6,:) = [0.95, -4, -3, -1.75, 0.25, 1.1];
rolltable(7,:) = [1.1, -7.1, -5.95, -4.3, -1.1, 0.8];
rolltable(8,:) = [1.2, -7, -5.5, -3.95, -1.25, 0.3];
rolltable(9,:) = [1.4, -8.9, -7.05, -5, -1.9, -0.1];
rolltable(10,:) = [1.5, -9, -7.2, -5.3, -2.1, -0.75];

% Table of values of heading (column1), pitch (column2), roll (column3) and speed
% (column4) for throughout the flight. Each row corresponds to a time after initial time.

flightmatrix = [0, 0, 0, 600;
                0, 0, 0, 600;
                0, 0, 0, 600;
                0, 0, 0, 600;
                0, 0, 0, 600;
                0, 0, 0, 600;
                0, 0, 0, 600;
                0, 0, 0, 600;
                0, 0, 0, 600;
                0, 0, 0, 600;
                0, 0, 0, 600;
                2.5,-2.5, 5, 600;
                5,-5, 10, 600;
                7.5,-7.5, 15, 600;
                10,-10, 20, 600;
                12.5,-12.5, 20, 600;
```

```
15,-15, 15, 600;
17.5,-17.5, 10, 600;
20,-20, 5, 600;
22.5,-20, 0, 600;
20,-17.5, -5, 600;
17.5,-15, -10, 600;
15,-12.5, -15, 600;
12.5,-10, -20, 600;
10,-7.5, -20, 600;
7.5,-5, -15, 600;
5,-2.5, -10, 600;
2.5, 0, -5, 600;
0, 0, 0, 600;
0, 0, 0, 600;
0, 0, 0, 600;];
```

```
% ****** LOCPOD.M ******
%
% The first section calculates each transmitters location in
% body frame coordinates. Since the body frame changes heading, pitch, and roll with
% aircraft, the only effects that need to be considered to generate the body frame
% coordinates are those of the pods pitch and roll, hence the formulas for each
% transmitters location. These are based on positive pitch being up and positive roll
% being up.

mach = spdplane/soundspeed;
altitude = downpos - radiusearth;
pitchpod = table2(pitchtable, mach, altitude)
rollpod = table2(rolltable, mach, altitude)

pitchpod=pitchpod*pi/180;    % need to convert these to radians for MATLAB use
rollpod=rollpod*pi/180;

% Transmitter 1 coordinates
        xbxmtr1 = 4.5*cos(pitchpod);
        ybxmtr1 = 15.5*cos(rollpod)-4.5*sin(rollpod)*sin(pitchpod);
        zbxmtr1 = -15.5*sin(rollpod)-4.5*sin(pitchpod)*cos(rollpod);

% Transmitter 2 coordinates
        xbxmtr2 = -4.5*cos(pitchpod);
        ybxmtr2 = 15.5*cos(rollpod)+4.5*sin(rollpod)*sin(pitchpod);
        zbxmtr2 = -15.5*sin(rollpod)+4.5*sin(pitchpod)*cos(rollpod);

% Transmitter 3 coordinates
        xbxmtr3 = 4.5*cos(pitchpod);
        ybxmtr3 = -15.5*cos(rollpod)+4.5*sin(rollpod)*sin(pitchpod);
        zbxmtr3 = -15.5*sin(rollpod)-4.5*sin(pitchpod)*cos(rollpod);

% Transmitter 4 coordinates
        xbxmtr4 = -4.5*cos(pitchpod);
        ybxmtr4 = -15.5*cos(rollpod)-4.5*sin(rollpod)*sin(pitchpod);
        zbxmtr4 = -15.5*sin(rollpod)+4.5*sin(pitchpod)*cos(rollpod);

% Convert these to NED:
[norxmtr1, eastxmtr1, downxmtr1] = body2nav(heading, pitch, roll, xbxmtr1, ybxmtr1,
                                                        zbxmtr1);
[norxmtr2, eastxmtr2, downxmtr2] = body2nav(heading, pitch, roll, xbxmtr2, ybxmtr2,
                                                        zbxmtr2);
[norxmtr3, eastxmtr3, downxmtr3] = body2nav(heading, pitch, roll, xbxmtr3, ybxmtr3,
```

```
[norxmtr3, eastxmtr3, downxmtr3] = body2nav(heading, pitch, roll, xbxmtr3, ybxmtr3,
                                                    zbxmtr3);
[norxmtr4, eastxmtr4, downxmtr4] = body2nav(heading, pitch, roll, xbxmtr4, ybxmtr4,
                                                    zbxmtr4);

[xxmtr1, yxmtr1, zxmtr1] =
        nav2ecef(norxmtr1,eastxmtr1,downxmtr1,norpos(time+1),eastpos(time+1));
[xxmtr2, yxmtr2, zxmtr2] =
        nav2ecef(norxmtr2,eastxmtr2,downxmtr2,norpos(time+1),eastpos(time+1));
[xxmtr3, yxmtr3, zxmtr3] =
        nav2ecef(norxmtr3,eastxmtr3,downxmtr3,norpos(time+1),eastpos(time+1));
[xxmtr4, yxmtr4, zxmtr4] =
        nav2ecef(norxmtr4,eastxmtr4,downxmtr4,norpos(time+1),eastpos(time+1));

actxmtr1=[xplane(time+1); yplane(time+1); zplane(time+1)]+[xxmtr1; yxmtr1; zxmtr1];
actxmtrx(1,time+1)=actxmtr1(1);
actxmtry(1,time+1)=actxmtr1(2);
actxmtrz(1,time+1)=actxmtr1(3);

actxmtr2= [xplane(time+1); yplane(time+1); zplane(time+1)]+[xxmtr2; yxmtr2; zxmtr2];
actxmtrx(2,time+1)=actxmtr2(1);
actxmtry(2,time+1)=actxmtr2(2);
actxmtrz(2,time+1)=actxmtr2(3);

actxmtr3= [xplane(time+1); yplane(time+1); zplane(time+1)]+[xxmtr3; yxmtr3; zxmtr3];
actxmtrx(3,time+1)=actxmtr3(1);
actxmtry(3,time+1)=actxmtr3(2);
actxmtrz(3,time+1)=actxmtr3(3);

actxmtr4= [xplane(time+1); yplane(time+1); zplane(time+1)]+[xxmtr4; yxmtr4; zxmtr4];
actxmtrx(4,time+1)=actxmtr4(1);
actxmtry(4,time+1)=actxmtr4(2);
actxmtrz(4,time+1)=actxmtr4(3);
```

```
% ******CPGPSSIM.M******
%
% This routine simulates the carrier phase range measurement.  It does
% so by calculating the actual range between the transmitters and receivers
% and then adding in the error terms for the carrier phase range equation.
% The user clock bias and transmitter clock biases are random numbers
% generated at the start and then used throughout the program.  Since these
% will be short duration simulations, no clock drift will be simulated, as
% the clock should not drift over a short time.  No ionospheric errors will
% be included as the plane will not be high enough for the ionosphere to
% affect the signal.  Phase ambiguity and cycle slips are not included since
% this research is not studying those particular problems.  The troposphere
% is modeled as constant throughout the simulation.


% variable descriptions
%
% trrngsq      : true range of transmitter to receiver squared; used for easier reading
% truerange    : true range between transmitter and receiver
% cprngmeas    : simulated carrier phase range measurement; includes error terms
%                note the triple indexing to account for 5 receivers for each transmitter at
%                each time
% transmitters : index for loop to account for 4 transmitters on aircraft & pseudolite
% receivers    : index for loop to account for the receivers in view

% Measurement simulation
for transmitters=1:4
% 4 transmitters on the aircraft; the pseudolite is included below
        for receivers=1:numrcvrs
        % allows for varying number of receivers - due to differencing, 5 or more
        % is best, with this research assuming 5, 4 for the solutions and 1 for
        % differencing
                trrngsq=(actxmtrx(transmitters,time+1)-xrcvr(receivers))^2
                        +(actxmtry(transmitters,time+1)-yrcvr(receivers))^2
                        +(actxmtrz(transmitters,time+1)-zrcvr(receivers))^2;
                truerange=sqrt(trrngsq);
                cprngmeas(transmitters,receivers)=truerange+rcvrclk(receivers)
                                                  +xmtrclk(transmitters)+tropdelay+randn;
                diffcorr=-rcvrclk(receivers)-xmtrclk(5)-tropdelay;
                diffmeas(transmitters, receivers)=
                                                cprngmeas(transmitters, receivers)+diffcorr;
        end
        findpos;
end
```

```
% ****** FINDPOS.M ******

% This is the least squares algorithm shown in Chapter 3.

stop=10;

vnom=[xposition(transmitters,time); yposition(transmitters,time);
                                        zposition(transmitters,time)];
vnom(4)=0;

while (stop>.000001)
        loops=loops+1;

        for i=1:numrcvrs
                phrc(i)=sqrt((vnom(1)-xrcvr(i))^2+(vnom(2)-yrcvr(i))^2
                        +(vnom(3)-zrcvr(i))^2)+vnom(4);
        end

        for i=1:numrcvrs
                h(i,1)=(vnom(1)-xrcvr(i))/(diffmeas(transmitters,i)-vnom(4));
                h(i,2)=(vnom(2)-yrcvr(i))/(diffmeas(transmitters,i)-vnom(4));
                h(i,3)=(vnom(3)-zrcvr(i))/(diffmeas(transmitters,i)-vnom(4));
                h(i,4)=1;
        end

        for i=1:numrcvrs
                dphr(i)=diffmeas(transmitters,i)-phrc(i);
        end

        dv=inv(h'*h)*h'*dphr';

        stop=sqrt(dv(1)^2+dv(2)^2+dv(3)^2+dv(4)^2);

        vnom=vnom+dv;
end

gdop(time+1)=sqrt(trace(inv(h'*h)));
xposition(transmitters, time+1)=vnom(1);
yposition(transmitters,time+1)=vnom(2);
zposition(transmitters, time+1)=vnom(3);
```

```
% ****** CHANGLOC.M ******
%
% This routine performs the actual movement of the aircraft from
% one position to the next.

% Find component speeds to make it easier to move the aircraft

downspd = sin(pitch*pi/180)*spdplane;
spdleft=sqrt(spdplane^2-downspd^2);
norspd = cos(heading*pi/180)*spdleft;
eastspd = sin(heading*pi/180)*spdleft;

% Actually move the aircraft. NOTE: This assumes one second updates.
% For quicker updates, multiply the speeds by the factor involved. Down
% is a simple addition of feet. For the north and east changes, it utilizes a
% function called FT2DEG.M so that the changes can be made easier. It
% is simpler to deal in NED until pod tips are located, then convert to ECEF
% to do rest of carrier-phase calculations easier.

downpos(time+1) = downpos(time)+downspd;
norpos(time+1) = norpos(time)+ft2deg(norspd, downpos(time+1));
eastpos(time+1) = eastpos(time)+ft2deg(eastspd, downpos(time+1));

[xplane(time+1), yplane(time+1), zplane(time+1)]=lla2ecef(norpos(time+1),
                                                    eastpos(time+1), downpos(time+1));
```

NOTE : THESE ARE THE FUNCTIONS USED BY THE REFERENCE MODEL

```
% ****** BODY2NAV.M ******
%
% This routine sets up the DCM to change from body frame to nav frame. It needs three
% parameters: heading, pitch, and roll. The DCM will then be created accordingly and
% the conversion will occur.
function [n,e,d] = body2nav(hdg, ptch, roll, xb, yb, zb)

% Set up the DCM
shdg = sin(hdg*pi/180);
chdg = cos(hdg*pi/180);
sptch = sin(ptch*pi/180);
cptch = cos(ptch*pi/180);
sroll = sin(roll*pi/180);
croll = cos(roll*pi/180);

DCM = [chdg*cptch, chdg*sptch*sroll-shdg*croll, chdg*sptch*croll+shdg*sroll;
       shdg*cptch, shdg*sptch*sroll+chdg*croll, shdg*sptch*croll-chdg*sroll;
         -sptch,cptch*sroll,cptch*croll];

bodypos = DCM * [xb;yb;zb];
n=bodypos(1);
e=bodypos(2);
d=bodypos(3);


% ****** NAV2ECEF.M ******
function [x,y,z]=nav2ecef(n, e, d, lat, long)

% Set up DCM

slat=sin(lat*pi/180);
clat=cos(lat*pi/180);
slong=sin(long*pi/180);
clong=cos(long*pi/180);

Cntoe=[-slat*clong, -slong, -clat*clong;
    -slat*slong, clong, -clat*slong;
       clat,      0,   -slat];

ecefpos=Cntoe*[n; e; d];
x=ecefpos(1);
y=ecefpos(2);
z=ecefpos(3);
```

```
% ******LLA2ECEF.M******
%
% This function converts latitude, longitude, and altitude to x,y,z coordinates
% in the ECEF frame. It assumes that the lat and long are given in DEGREES and
% that the earth is spherical.

function [x,y,z] = lla2ecef(lat,long,alt)

lat=lat*pi/180;
long=long*pi/180;
% These are needed to convert to azimuth, elevation and radius
[x,y,z]=sph2cart(long, lat, alt);


% ******ECEF2LLA.M******
%
% This function converts x,y,z coordinates in the ECEF frame to latitude,
% longitude, and altitude (from the center of the earth). It assumes that
% the earth is spherical. North and East measures are positive.

function [lat,long,alt] = ecef2lla(x,y,z)

[long,lat,alt]=cart2sph(x,y,z);
lat=lat*180/pi;
long=long*180/pi;
% These are needed to convert from azimuth, elevation and radius
```

```
% ****** FT2DEG.M ******
% This function uses the speed of the aircraft in component directions to
% determine how much the latitude or longitude changes at each update time.
% IMPORTANT: this is based on the spherical earth model, therefore one
% function can do it for latitude and longitude. If updated for WGS-84 use
% it will be necessary to write separate routines for each.
% It utilizes the standard trig relationship that ratio of the length of an
% arc to the circumference is equal to the ratio of the angle of arc to
% 2pi, 360 degrees.
%
% Variables used:
% circumearth : circumference of the earth, including the altitude of the aircraft
% speed       : dummy variable that will equal the speed of whatever the
%                 component direction being considered is

function chng = ft2deg(speed, down);

circumearth = 2*(down)*pi;
chng = 360*speed/circumearth;
```

```
% ****** FINDANP.M ******

% This routine calculates the attitude and position of the aircraft given the transmitter
% locations.

for trial = 1:numtrials
        for t=2:(endsim+1)
                approxspd=sqrt((xmtr1x(trial,t)-xmtr1x(trial,t-1))^2
                                +(xmtr1y(trial,t)-xmtr1y(trial,t-1))^2
                                +(xmtr1z(trial,t)-xmtr1z(trial,t-1))^2);
                [xmtr1lat(t), xmtr1long(t), xmtr1down(t)]=
                                ecef2lla(xmtr1x(trial,t), xmtr1y(trial,t), xmtr1z(trial,t));
                [xmtr2lat(t), xmtr2long(t), xmtr2down(t)]=
                                ecef2lla(xmtr2x(trial,t), xmtr2y(trial,t), xmtr2z(trial,t));
                avglat(t)=(xmtr1lat(t)+xmtr2lat(t))/2;
                avglong(t)=(xmtr1long(t)+xmtr2long(t))/2;
                [navx1(t), navy1(t), navz1(t)]=ecef2nav(xmtr1x(trial,t), xmtr1y(trial,t),
                                                xmtr1z(trial,t), avglat(t), avglong(t));
                [navx2(t), navy2(t), navz2(t)]=ecef2nav(xmtr2x(trial,t), xmtr2y(trial,t),
                                                xmtr2z(trial,t), avglat(t), avglong(t));

                [xmtr3lat(t), xmtr3long(t), xmtr3down(t)]=
                                ecef2lla(xmtr3x(trial,t), xmtr3y(trial,t), xmtr3z(trial,t));
                [xmtr4lat(t), xmtr4long(t), xmtr4down(t)]=
                                ecef2lla(xmtr4x(trial,t), xmtr4y(trial,t), xmtr4z(trial,t));

                approxmach=approxspd/soundspeed;
                approxalt=(pod1down(t)-radiusearth);
                approxpodptch=table2(pitchtable, approxmach, approxalt);
                approxpodroll=table2(rolltable, approxmach, approxalt);

                dist1to3=sqrt((xmtr1x(trial,t)-xmtr3x(trial,t))^2
                                +(xmtr1y(trial,t)-xmtr3y(trial,t))^2
                                +(xmtr1z(trial,t)-xmtr3z(trial,t))^2);
                chngroll=1;
                oldroll=0;
                while (chngroll>0.01)
                        pitcharg=(pod1down(t)-pod2down(t))/9;
                        if abs(pitcharg)>1
                                aircraftpitch(trial,t)=sign(pitcharg)*90;
                        else
                                aircraftpitch(trial,t)=asin(pitcharg)*180/pi-
                                        approxpodptch*cos((oldroll-approxpodroll)*pi/180);
                        end;
```

```
            if cos(aircraftpitch(trial,t)*pi/180)~=0
                rollarg=(pod3down(t)-pod1down(t))/
                    (dist1to3*cos(aircraftpitch(trial,t)*pi/180));
                if (abs(rollarg)<1)
                    if (pod1long(t)>pod3long(t))
                        newroll=asin(rollarg)*180/pi;
                    else
                        newroll=sign(rollarg)*
                            (abs(asin(rollarg)*180/pi)+90);
                    end
                else
                    if (pod3down(t)>pod1down(t))
                        newroll=90;
                    else
                        newroll=-90;
                    end
                end
            else
                if (pod3down(t)>pod1down(t))
                    newroll=90;
                else
                    newroll=-90;
                end
            end
        chngroll=oldroll-newroll;
        oldroll=newroll;
        end             % while loop end
        aircraftroll(trial,t)=oldroll;

        if (navx1(t)-navx2(t))~=0
            hdgarg=(navy1(t)-navy2(t))/(navx1(t)-navx2(t));
            if (pod1lat(t)>pod2lat(t))
                airhdg(trial,t)=atan(hdgarg)*180/pi;
            else
                airhdg(trial,t)=sign(hdgarg)*(abs(atan(hdgarg)*180/pi)+90);
            end

beta1=4.5*cos(approxpodptch*pi/180)*cos(aircraftpitch(trial,t)*pi/180)
      +4.5*sin(approxpodptch*pi/180)*sin(aircraftpitch(trial,t)*pi/180)
            *cos(aircraftroll(trial,t)*pi/180);
beta2=4.5*sin(approxpodptch*pi/180)*sin(aircraftroll(trial,t)*pi/180);
            aircrafthdg(trial,t)=atan((-beta1*tan(airhdg(trial,t)*pi/180)-beta2)/
                (beta2*tan(airhdg(trial,t)*pi/180)-beta1))*180/pi;
        else
```

```
                if (pod1long(t)>pod2long(t))
                        airhdg(trial,t)=90;
                else
                        airhdg(trial,t)=-90;
                end
        end

    xaircraftpos(trial,t)=(xmtr1x(trial,t)+xmtr2x(trial,t)+xmtr3x(trial,t)+xmtr4x(trial,t))
                                /4;
    yaircraftpos(trial,t)=(xmtr1y(trial,t)+xmtr2y(trial,t)+xmtr3y(trial,t)+xmtr4y(trial,t))
                                /4;
    zaircraftpos(trial,t)=(xmtr1z(trial,t)+xmtr2z(trial,t)+xmtr3z(trial,t)+xmtr4z(trial,t))
                                /4;
    [lat(t), long(t), alt(t)]=
        ecef2lla(xaircraftpos(trial,t), yaircraftpos(trial,t), zaircraftpos(trial,t));
    adj=.5*(sqrt((xmtr1x(trial,t)-xmtr3x(trial,t))^2+(xmtr1y(trial,t)-xmtr3y(trial,t))^2
        +(xmtr1z(trial,t)-xmtr3z(trial,t))^2))*tan(approxpodroll*pi/180);
    [ladj, loadj, dadj]=
        body2nav(aircrafthdg(trial,t),aircraftpitch(trial,t),aircraftroll(trial,t),0,0,adj);
    [xadj, yadj, zadj]=nav2ecef(ladj, loadj, dadj, lat(t), long(t));
    xaircraftpos(trial,t)=xaircraftpos(trial,t)-xadj;
    yaircraftpos(trial,t)=yaircraftpos(trial,t)-yadj;
    zaircraftpos(trial,t)=zaircraftpos(trial,t)-zadj;
    [aircraftlat(trial,t), aircraftlong(trial,t), aircraftdown(trial,t)]=
                ecef2lla(xaircraftpos(trial,t),yaircraftpos(trial,t),zaircraftpos(trial,t));
    end
end
```

```
% ******ECEF2LLA.M******
%
% This function converts x,y,z coordinates in the ECEF frame to latitude,
% longitude, and altitude (from the center of the earth). It assumes that
% the earth is spherical. North and East measures are positive.

function [lat,long,alt] = ecef2lla(x,y,z)

[long,lat,alt]=cart2sph(x,y,z);
lat=lat*180/pi;
long=long*180/pi;
% These are needed to convert from azimuth, elevation and radius



% ****** ECEF2NAV.M ******

function [n,e,d]=ecef2nav(x,y,z, lat, long)

% Set up DCM

slat=sin(lat*pi/180);
clat=cos(lat*pi/180);
slong=sin(long*pi/180);
clong=cos(long*pi/180);

Cntoe=[-slat*clong, -slong, -clat*clong;
     -slat*slong, clong, -clat*slong;
        clat,      0,   -slat];

Ceton=Cntoe';
navpos=Ceton*[x;y;z];
n=navpos(1);
e=navpos(2);
d=navpos(3);
```

Appendix B. Results of Flight Profile 1

This appendix contains the transmitter location results not provided in Chapter 4 for the reader. The first three plots are transmitter position error for transmitters two through four of the four-receiver case. These are followed by plots of the same errors for the five-receiver case.

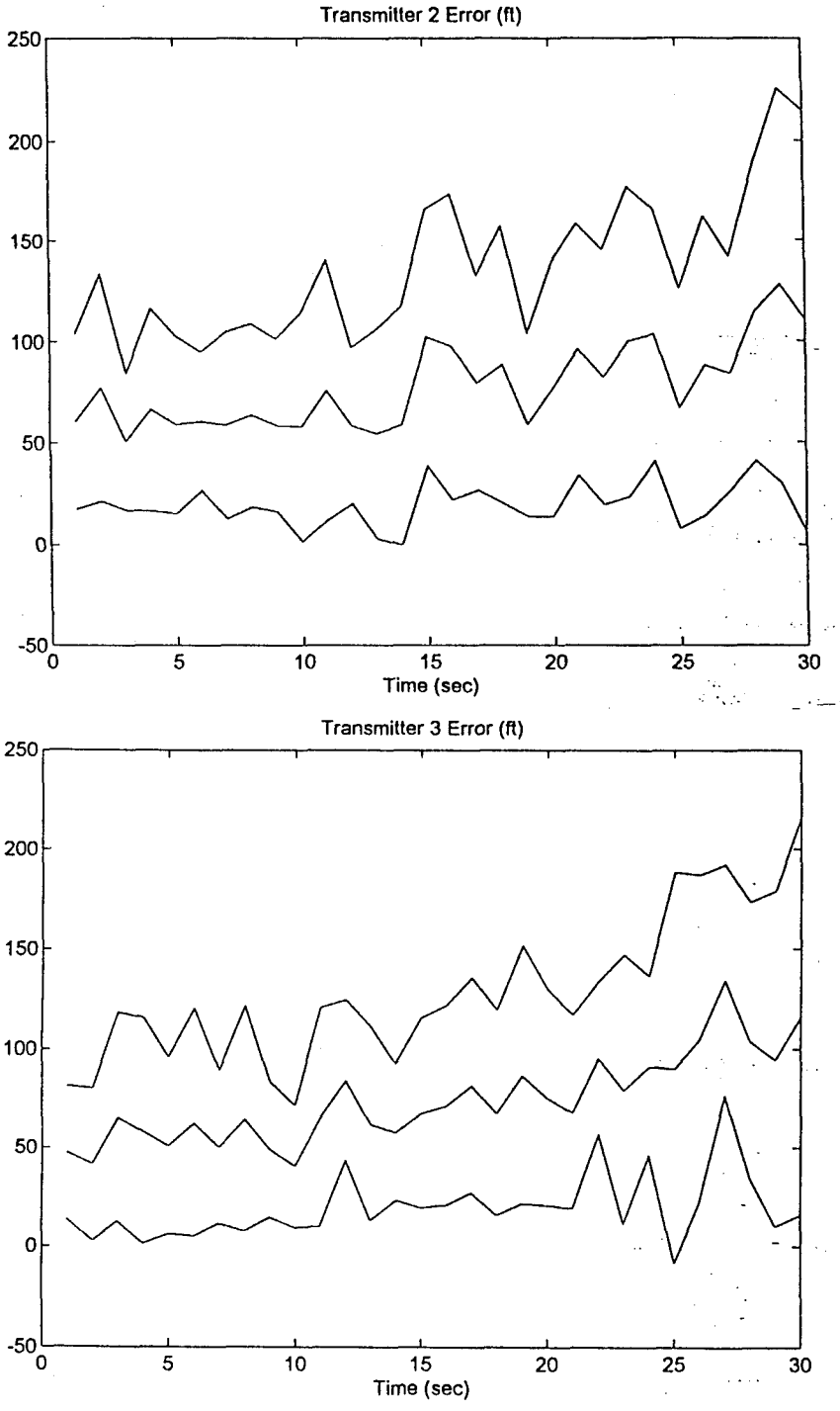

Figure B.1 Transmitter 2 and Transmitter 3 Position Error
Flight Profile 1, Four-Receiver Case

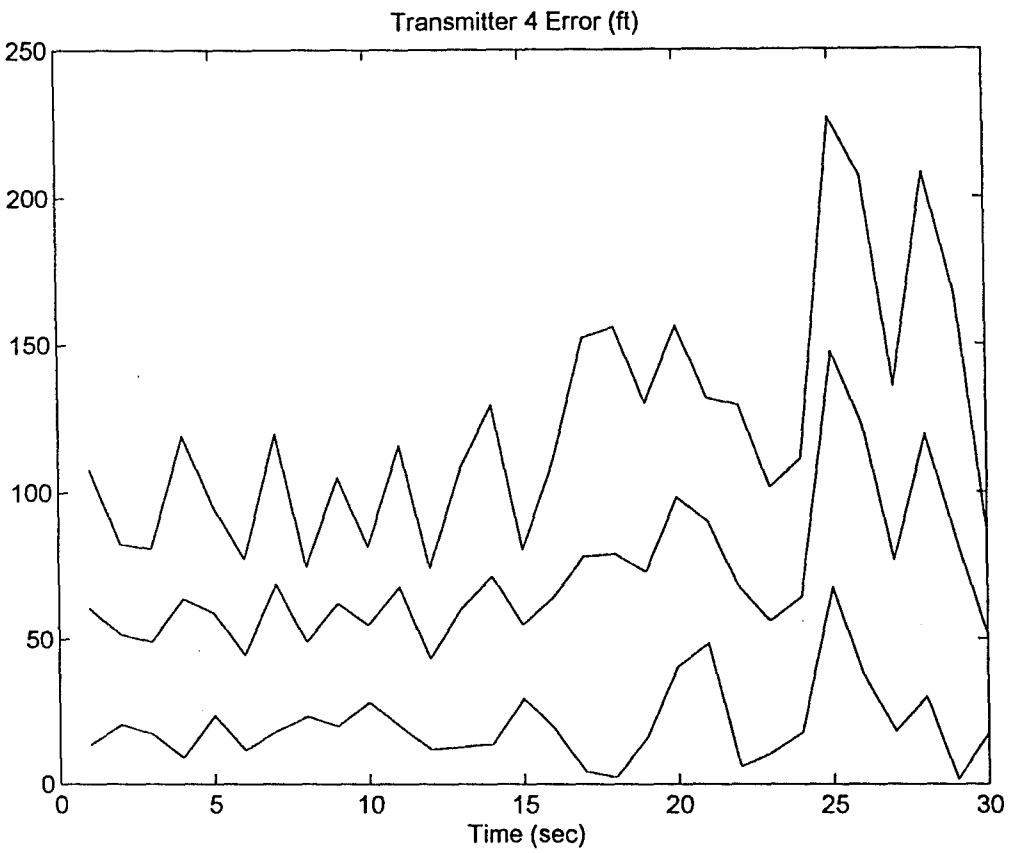

Figure B.2 Transmitter 4 Position Error
Flight Profile 1, Four-Receiver Case

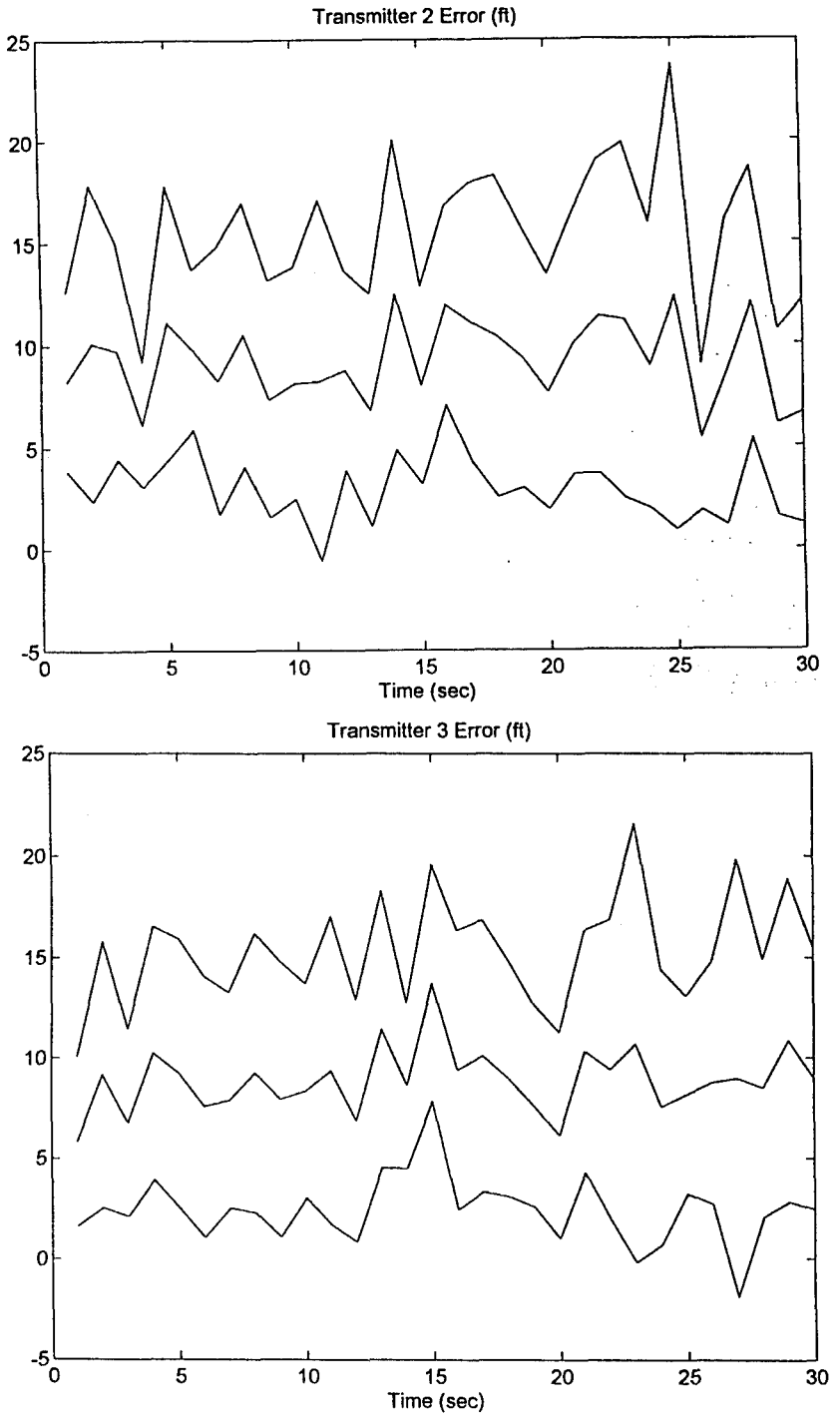

Figure B.3 Transmitter 2 and Transmitter 3 Position Error
Flight Profile 1, Five-Receiver Case

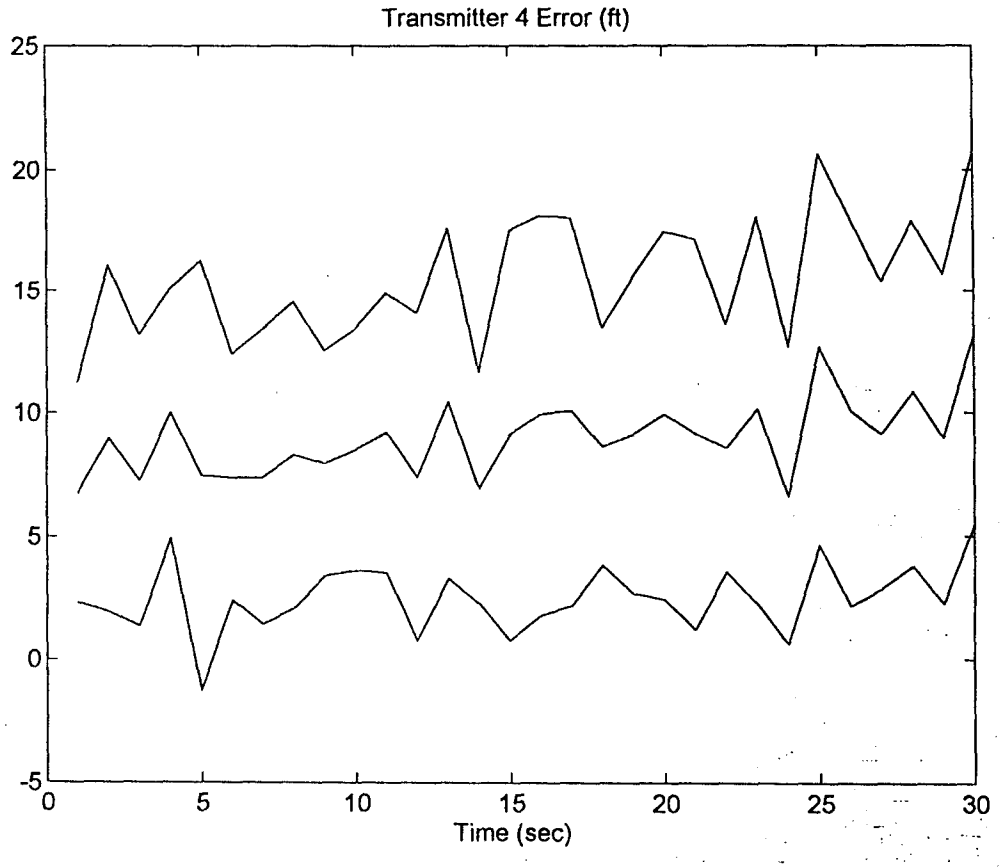

Figure B.4 Transmitter 4 Position Error
Flight Profile 1, Five-Receiver Case

Appendix C. Results of Flight Profile 2

This appendix contains the transmitter location results not provided in Chapter 4 for the reader. The first three plots are transmitter position error for transmitters two through four of the five-receiver case. These are followed by plots of the attitude errors utilizing Dr. van Graas' attitude determination technique in this situation.

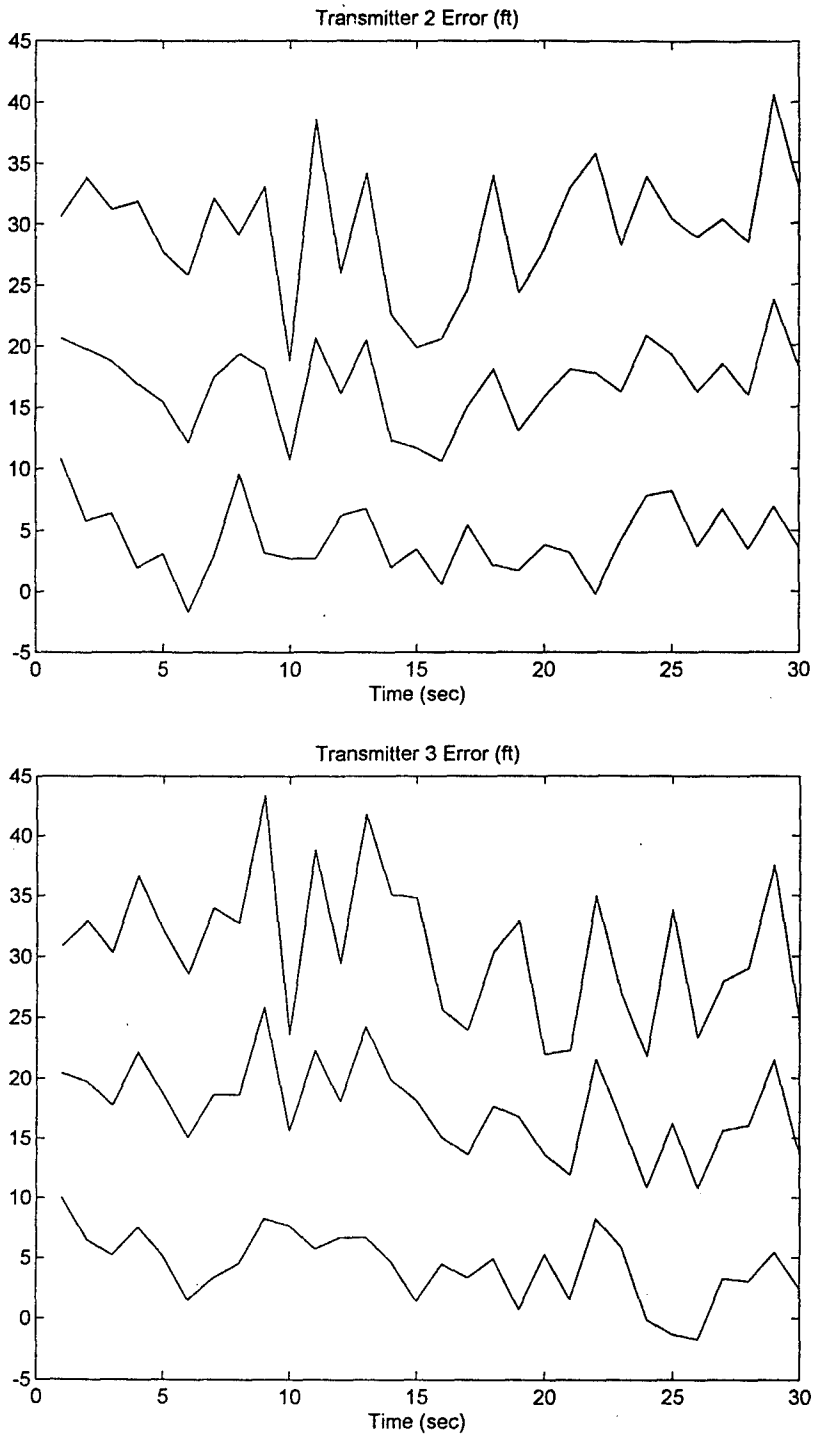

Figure C.1 Transmitter 2 and Transmitter 3 Position Error
Flight Profile 2, Five-Receiver Case

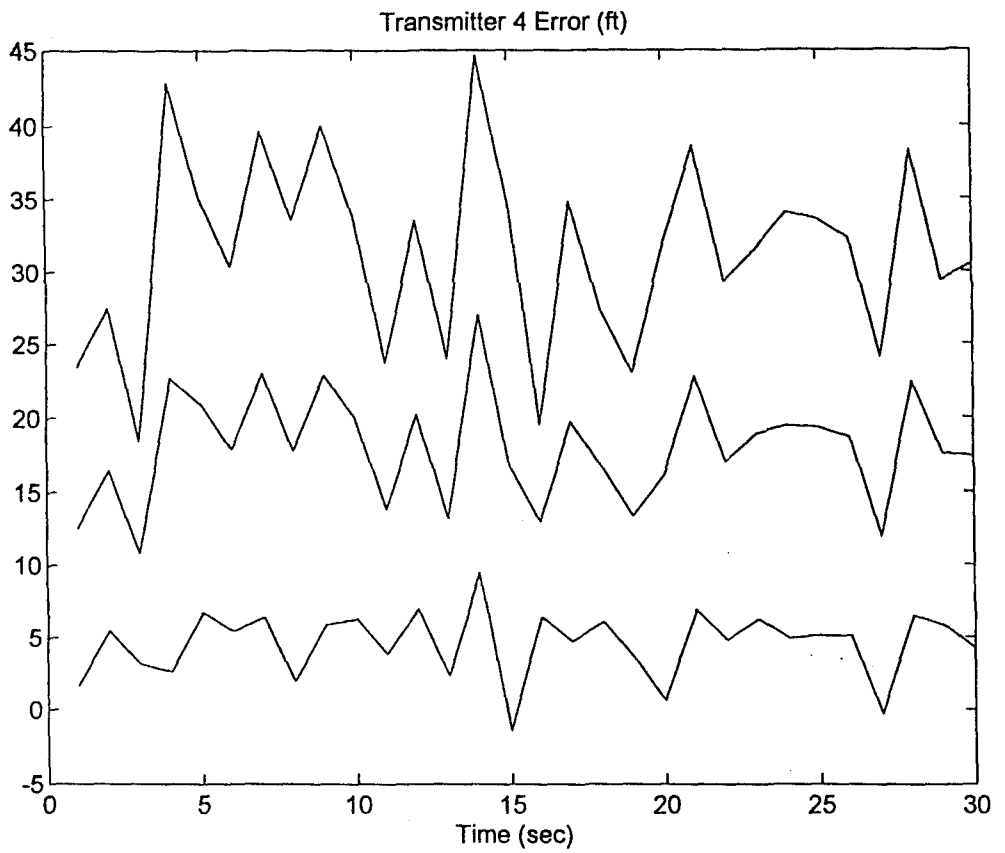

Figure C.2 Transmitter 4 Position Error
Flight Profile 2, Five-Receiver Case

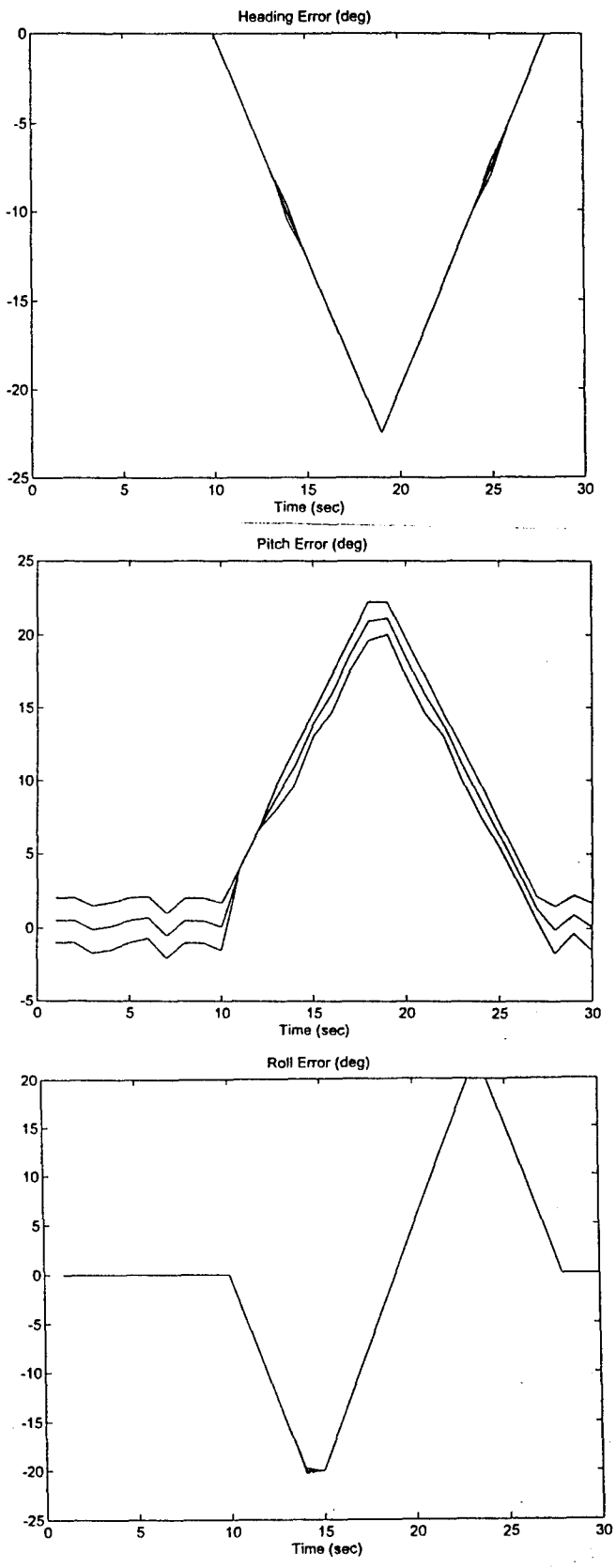

Figure C.3 Attitude Errors - Dr. van Graas' Method

Bibliography

1. Bohenek, Brian J. *The Enhanced Performance of an Integrated Navigation System in a Highly Dynamic Environment.* MS Degree Thesis, AFIT/GE/ENG/94D-01. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1994.

2. Hwang, Patrick Y.C. and R. Grover Brown. "GPS Navigation: Combining Pseudorange with Continuous Carrier Phase Using a Kalman Filter," *Global Positioning System.* Papers Published in *Navigation,* Volume IV:175-190, The Institute of Navigation, Alexandria, VA 22314, 1993.

3. Nash, Captain Tony. 746th Test Squadron, Holloman AFB, NM. Personal Interview. 16 June 1995.

4. van Graas, Frank, David W. Diggle and Richard M. Hueschen. "Interferometric GPS Flight Reference/Autoland System: Flight Test Results," *Navigation, 41*: 57-82 (Spring 1994).

5. Cohen, Clark E., Bradford W. Parkinson and B. David McNally. "Flight Tests of Attitude Determination Using GPS Compared Against an Inertial Navigation Unit," *Navigation, 41*: 83-98 (Spring 1994).

6. Garfinkle, Moishe. "Twisting Smartly In The Wind," *Aerospace America,* July 1994: 18-20.

7. Shelton, D. A. *F-16 C/D Block 40 Aircraft Wing Twist Analysis Report.* Technical Report, General Dynamics, July 1988. Contract F33657-82-C-2038: CDRL 10019.

8. Remondi, Benjamin W. "Performing Centimeter-Level Surveys in Seconds With GPS Carrier Phase: Initial Results," *Global Positioning System.* Papers Published in *Navigation,* Volume III:194-208, The Institute of Navigation, Alexandria, VA 22314, 1993.

9. Angelucci, Enzo. *The Rand McNally Encyclopedia of Military Aircraft: 1914 to the Present.* New York: Crescent Books, 1990.

10. Krabill, William B. and Chreston F. Martin. "Aircraft Positioning Using Global Positioning Carrier Phase Data," *Global Positioning System.* Papers Published in *Navigation,* Volume IV:299-320, The Institute of Navigation, Alexandria, VA 22314, 1993.

11. Canadian GPS Associates. *Guide to GPS Positioning.* ISBN: 0-920-114-73-3, Fredericton, New Brunswick: University of New Brunswick Graphic Services, May 1987.

12. Hansen, Neil P. *Incorporation of Carrier Phase Global Positioning System Measurements into the Navigation Reference System for Improved Performance.* MS Degree Thesis, AFIT/GE/ENG/93D-40. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1993.

13. Zapata, Faustino. F-16 System Program Office, Structures Division, Wright-Patterson AFB OH. Personal Interview. 23 June 1995.

14. Hsu, Hwei P. *Applied Vector Analysis.* Orlando, Florida: Harcourt Brace Jovanovich, 1984.

15. World Geodesic Survey, 1984.

16. *Matlab.* Version 4.1. Computer Software. The MathWorks, Inc. 24 Prime Park Way, Natick, MA 01760. June 1993.

17. Anderson, John D. *Fundamentals of Aerodynamics, 2nd ed.* New York: McGraw-Hill, Inc., 1991.

18. Lewantowicz, Colonel Zdzislaw H. *Fundamentals of Aerospace Instruments and Navigation Systems: Course Notes for EENG 534.* Department of Electrical and Computer Engineering, Air Force Institute of Technology. October 1989.

19. Delap, Captain Ronald. Assistant Professor of Electrical Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH. Course Notes EENG 534. Fall 1994.

20. Blackwell, Earl G. "Overview of Differential GPS Methods," *Global Positioning System.* Papers Published in *Navigation,* Volume III:89-100, The Institute of Navigation, Alexandria, VA 22314, 1993.

21. Riggins, Lt. Colonel Robert N. Assistant Professor of Electrical Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH. Course Notes EENG 744. Winter 1995.

22. Noe, P. S., K. A. Myers and T. K. Wu. "A Navigation Algorithm for the Low-Cost GPS Receiver," *Global Positioning System.* Papers Published in *Navigation,* Volume I:166-172, The Institute of Navigation, Alexandria, VA 22314, 1993.

23. van Graas, Frank and Michael Braasch. "GPS Interferometric Attitude and Heading Determination: Initial Flight Test Results," *Global Positioning System.* Papers Published in *Navigation,* Volume IV:359-377, The Institute of Navigation, Alexandria, VA 22314, 1993.

Vita

Second Lieutenant Bradley W. Mahlum [redacted]. After many moves and graduation from Bloomingdale Senior High School, Valrico, Florida, Bradley enrolled in the University of Notre Dame as a member of the Air Force Reserve Officer Training Corps (ROTC) detachment. In May 1994, Bradley graduated from the University of Notre Dame with a BSEE and was called on active duty to enter the Air Force Institute of Technology (AFIT), Wright-Patterson AFB, Ohio. At AFIT, he earned a Masters of Science in Electrical Engineering with a concentration in navigation systems. After graduating from AFIT, Bradley was assigned to the Avionics Lab, Wright-Patterson AFB, Ohio. While assigned here, he worked with integration of navigation systems and planned his wedding to his fiancee, Aimee.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1995	Master's Thesis

4. TITLE AND SUBTITLE

Wing Flexure Compensation For Attitude and Position Determination in an Inverted Carrier-Phase Positioning System

5. FUNDING NUMBERS

6. AUTHOR(S)

Bradley W. Mahlum

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/95D-14

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Capt. Tony Nash
746th Test Squadron
Holloman AFB, NM

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In response to the ever increasing accuracies in inertial navigation systems, the U. S. Air Force must develop higher accuracy reference systems. These reference systems must also be small enough to be utilized in the testing of navigation systems onboard fighter aircraft. One such proposed system utilizes Carrier-Phase Global Positioning System (CPGPS) transmitters mounted on AIM-9 pods with receivers on the ground. This research examines one possible method of utilizing this system to determine the attitude and position of the aircraft, given position estimates for transmitter's locations. The transmitter positioning algorithm showed that the geometry will be problematic for this configuration. However, if given the estimates of transmitter positions within the desired accuracy, the aircraft attitude and position algorithms worked effectively.

14. SUBJECT TERMS

15. NUMBER OF PAGES
101

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL